

Střední průmyslová škola Třebíč

Studijní obor: Informační technologie

Třída: ITA4

Školní rok: 2023/2024 Lukáš Kurtin

Maturitní práce

Simulační program Rubikovy kostky

Profilová část maturitní zkoušky

Zadání práce

ABSTRAKT

Tvorba maturitní práce je jedním z velmi klíčových momentů při studiu. Kvalita zpracování její formální části je pak jedním z nejdůležitějších kritérií při jejím hodnocení. Cíl této práce je popsat jednotlivé kroky během tohoto procesu, doporučit postupy a vytvořit šablonu, která usnadní celý proces.

KLÍČOVÁ SLOVA

maturitní práce, šablona

ABSTRACT

The creation of a graduation thesis is one of the most crucial moments during studies. The quality of the processing of its formal part is then one of the most important criteria in its evaluation. The aim of this work is to describe the individual steps during this process, recommend procedures and create a template that will facilitate the entire process.

KEYWORDS

graduation thesis, template

PODĚKOVÁNÍ

Děkuji Natálii Pistovčákové, která nám pomáhá nepropadnout za plagiát.

V Třebíči dne 22. ledna 2024 podpis autora

PROHLÁŠENÍ

Prohlašuji, že jsem tuto práci vypracoval/a samostatně a uvedl/a v ní všechny prameny, literaturu a ostatní zdroje, které jsem použil/a.

V Třebíči dne 22. ledna 2024

podpis autora

Obsah

[Úvod 6](#_Toc151068106)

[1 Teoretická část 7](#_Toc151068107)

[1.1 Rubikova kostka 7](#_Toc151068108)

[1.2 Algoritmizace 7](#_Toc151068109)

[1.2.1 Značení tahů 7](#_Toc151068110)

[1.3 Programová struktura kostky 8](#_Toc151068111)

[1.3.1 Způsob 1 8](#_Toc151068112)

[1.3.2 Způsob 2 8](#_Toc151068113)

[1.4 Grafické vykreslení 8](#_Toc151068114)

[1.5 Použité technologie 9](#_Toc151068115)

[1.5.1 GitHub 9](#_Toc151068116)

[1.5.2 Windows Forms .NET Framework 9](#_Toc151068117)

[2 Praktická část 10](#_Toc151068118)

[Závěr 11](#_Toc151068119)

[2.1 Psaní úvodu 11](#_Toc151068120)

[2.2 Struktura odstavců 11](#_Toc151068121)

[2.3 Obrázky, tabulky a rovnice 12](#_Toc151068122)

[2.4 Řazení a struktura kapitol 14](#_Toc151068123)

[2.5 Závěr 14](#_Toc151068124)

[2.6 Seznam použitých zdrojů 14](#_Toc151068125)

[2.7 Seznam použitých symbolů a zkratek 14](#_Toc151068126)

[2.8 Seznamy použitých obrázků a tabulek 15](#_Toc151068127)

[2.9 Seznam příloh 15](#_Toc151068128)

[Závěr 16](#_Toc151068129)

[Seznam použitých zdrojů 17](#_Toc151068130)

[Seznam použitých symbolů a zkratek 18](#_Toc151068131)

[Seznam obrázků 19](#_Toc151068132)

[Seznam tabulek 20](#_Toc151068133)

[Seznam příloh 21](#_Toc151068134)

Úvod

Jako moji maturitní ročníkovou práci jsem chtěl vypracovat projekt nad moje zkušenosti, u kterého bych strávil více času než na jakémkoliv jiném projektu, který jsem v průběhu školních let dělal. Ideální je proto projekt, který je náročný nejen z hlediska programátorského, ale i grafického.

Vybral jsem si proto simulaci Rubikovy kostky, které jsou mi již léta blízké a vždy mě fascinovala jejich funkčnost. Pro uskutečnění bude potřeba projít několik způsobů, jak naprogramovat základní strukturu a funkce kostky.

Graficky bude kostka zobrazena v krychlové síti a ve 3D prostoru s možností otáčení pohledu. K tomu bude zapotřebí nastudovat projekční matice, které umožňují překlad mezi 3D prostorem a 2D obrazovkou.

Jako vývojové prostředí použiji Microsoft Visual studio a jeho balíček Windows Forms .NET framework, ve kterém se programuje v jazyku C# a který ulehčuje vykreslování na obrazovku a zadávání uživatelského vstupu. Pro řízení projektu a správu verzí využiji platformu GitHub, více popsanou v teoretické části.

Konečný stav projektu si představuji jako intuitivní aplikaci, která bude obsahovat samotné funkce Rubikovy kostky, ale i teoretický návod na její složení pomocí nejpoužívanější metody CFOP (pojmenované podle čtyř kroků, které aplikuje). Také bych chtěl implementovat optimalizovaný algoritmus pro automatické složení kostky, popřípadě ho využít jako doplněk naučné stránky.

# Teoretická část

Přidat popis teoretické části

## Rubikova kostka

Velký počet lidí si o Rubikově kostce myslí, že se skládá pomocí zaměňování barev, což není plně pravda. Skutečně se totiž neotáčí s barvami, ale s jednotlivými kostičkami. Kostka jich má celkově 22, 8 rohů se třemi barvami a 12 hran se dvěma. Jednobarevné středy se do nich nezapočítávají, neboť jejich poloha se nikdy nemění. Jsou totiž středem osy každého otočení.

Každá kostička má distinktivní kombinaci barev, které za normálních okolností nemohou mít duplikát. Může se měnit pouze jejich poloha a tím i její otočení.

## Algoritmizace

Jedním z prvků finální verze projektu by měl být algoritmus pro metodické složení kostky. Pokud se kostkou bude náhodně otáčet, složení by zabralo průměrně 43,252×1015 tahů. Existuje proto několik algoritmických sad, ze kterých se vytváří skládací metody. Nejznámějšími a nejpoužívanějšími z nich jsou CFOP[4] a ROUX. Kvůli svojí znalosti o této metodice použiji CFOP, dále popsanou v praktické části.

Jako zdroj algoritmů využiji veřejnou databázi algdb.net. [2]

### Značení tahů

Algoritmy se zapisují podle celosvětové notace, kde každý tah má svoje písmeno z anglického názvu. Základní jsou R, L, U, D, F, B (pravá, levá, horní, spodní, přední, zadní). Otáčí se po směru hodinových ručiček. Rozšiřující jsou pak široké tahy w, rotace pohledu na kostku x, y, z a průřezové M, E, S. Každý tah se také řídí pravidlem, že pokud se za písmenem nachází apostrof, tah se provede v opačném směru a pokud číslice 2, provede se posun 2x.[1]

## Programová struktura kostky

Velmi důležitý faktor je, jak vytvořit algoritmus, který bude kostkou otáčet. Na základě toho se vytvoří celý systém a struktura kostky. Po troše hledání jsem došel ke dvěma možným způsobům. Kvůli převažujícím výhodám využiji způsob 2.

### Způsob 1

Síla tohoto způsobu je nízká náročnost programovatelnosti.

Kostka by se skládala z 6 polí o dvou dimenzích neboli 3x3 matice. Každý prvek by znamenal jedno barevné pole na kostce.

Spočívá v tom, že při zadání příkazu na otočení strany se každá barva na každé ovlivněné kostičce natvrdo posune na její požadovanou pozici. Nevýhodou je neaplikovatelnost v chytrých algoritmech, neboť by se při nich musely samostatně kontrolovat všechny barvy. To by začinilo extrémní časovou náročnost programu  
a zbytečně dlouhý a nepřehledný kód.

### Způsob 2

Další varianta je těžší na programování, ale je kompatibilnější s dynamickým vývojem.

Namísto šesti matic se využijí pouze 3, jedna na každou vrstvu kostky. Jejich objekty by byly třídy kostičky, které by obsahovaly vlastnosti otočení a pevné očíslování základní pozice. Kostičky by měnily svoji aktuální polohu po směru hodinových ručiček relativně k ostatním. Pro získávání stavů kostky pro užití vhodného skládacího algoritmu by se kontrolovala pouze rotace a očíslování kostičky (vztahy mezi větším počtem kostiček).

## Grafické vykreslení

Pro vykreslení kostky se použijí 2 způsoby: síťové zobrazení kostky pro 2D  
a ortodoxní projekce pro 3D.

### Ortodoxní projekce

## Použité technologie

### GitHub

GitHub je online služba, která poskytuje ukládání projektů a jejich přenos mezi zařízeními. Dělá se tak pomocí repositářů.

#### GitHub Desktop

Aplikace, která ulehčí práci s platformou GitHub.

### Windows Forms .NET Framework

# Praktická část

Závěr

Seznam použitých zdrojů

[1] [Rubik's Cube Move Notation (jperm.net)](https://jperm.net/3x3/moves)

[2] [3x3 (algdb.net)](http://algdb.net/puzzle/333)

[3] <https://www.youtube.com/watch?v=ih20l3pJoeU>

[4] [Rubik's Cube solution with advanced Fridrich (CFOP) method (ruwix.com)](https://ruwix.com/the-rubiks-cube/advanced-cfop-fridrich/)

Seznam použitých symbolů a zkratek

- předělat

|  |  |  |
| --- | --- | --- |
| Zkratka | Celý název |  |
| CFOP | Kříž F2L OLL PLL |  |
| F2L | První 2 vrstvy |  |
| OLL | Orientace poslední vrstvy |  |
| PLL | Permutace poslední vrstvy |  |
|  |  |  |
|  |  |  |

Seznam obrázků

[Obr. 2.1 Obsah 11](#_Toc147493921)

[Obr. 2.2 Příklad umístění legendy obrázku 13](#_Toc147493922)

Seznam tabulek

[Tab. 2.1 Legenda k tabulce 13](#_Toc147493615)

Seznam příloh

Prázdná šablona maturitní práce