


Řešení Rubikovy kostky pomocí kolaborativního robota se dvěma pažemi

Matej Krupčiak

Bakalářská práce
2020

 Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

Ústav automatizace a řídicí techniky

Akademický rok: 2019/2020

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Matej Krupčiak**
Osobní číslo: **A17040**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Inteligentní systémy s roboty**
Forma studia: **Prezenční**
Téma práce: **Řešení Rubikovy kostky pomocí kolaborativního robota se dvěma pažemi**
Téma práce anglicky: **Rubik Cube Solutions Using a Dual-Arm Collaborative Robot**

Zásady pro vypracování

1. Vypracujte literární rešerši na dané téma.
2. Připravte v prostředí RobotStudio podklady ve formě modelů a mechanismů.
3. Implementujte algoritmus řešení Rubikovy kostky do programovacího jazyka RAPID.
4. Celý systém propojte do funkčního celku a otestujte.
5. Navrhněte způsob komunikace a přípravu k využití výsledků práce jako digitálního dvojčete k reálnému systému.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. *The Mathematics of the Rubik's Cube: Introduction to Group Theory and Permutation Puzzles* [online]. 2009 [cit. 2019-11-17]. Dostupné z: <http://web.mit.edu/sp.268/www/rubik.pdf>. Massachusetts Institute of Technology
2. ROKICKI, Tomas, Herbert KOCIEMBA, Morley DAVIDSON a John DETHRIDGE. The Diameter of the Rubik's Cube Group Is Twenty. *SIAM Journal on Discrete Mathematics*. 2013, 27(2), 1082-1105. DOI: 10.1137/120867366. ISSN 0895-4801. Dostupné také z: <http://epubs.siam.org/doi/10.1137/120867366>
3. ZENG, Da-Xing, Ming LI, Juan-Juan WANG, Yu-Lei HOU, Wen-Juan LU a Zhen HUANG. Overview of Rubik's Cube and Reflections on Its Application in Mechanism. *Chinese Journal of Mechanical Engineering*. 2018, 31(1). DOI: 10.1186/s10033-018-0269-7. ISSN 1000-9345. Dostupné také z: <https://cjme.springeropen.com/artides/10.1186/s10033-018-0269-7>
4. *RobotStudio Operating Manual*. 3HAC032104-001. ABB, 2019
5. *Product Manual – IRB 14000*. 3HAC052983-001. ABB, 2018

Vedoucí bakalářské práce:

Ing. Luboš Spaček
Ústav řízení procesů

Datum zadání bakalářské práce: 20. prosince 2019
Termín odevzdání bakalářské práce: 15. května 2020



doc. Mgr. Milan Adámek, Ph.D.
děkan

prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

Jméno, příjmení: MATEJ KRUPČIAK

Název bakalářské/diplomové práce: Řešení Rubikovy kostky pomocí kolaborativního robota se dvěma pažemi

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užit své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové/bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne

Matej Krupčiak, v.r.....
podpis diplomanta

ABSTRAKT

Abstrakt slovensky

Táto bakalárska práca sa zaoberá riešením Rubikovej kocky za pomoci kolaboratívneho dvojramenného robota od firmy ABB, pod označením IRB 14000 YuMi. Cieľom práce je prostredníctvom robota dosiahnuť riešenie Rubikovej kocky za použitia rovnakého algoritmu, ktorým by ju riešil človek, v kontraste s použitím algoritmu generovaného počítačom. Práca pojednáva o histórii Rubikovej kocky, a zároveň aj o iných existujúcich algoritmoch a prináša hlbšie vysvetlenie použitého algoritmu, ale aj jeho implementáciu do programovacieho jazyka RAPID, využitie nástrojov vývojového prostredia RobotStudio pre vizualizáciu skladania a v neposlednom rade prepojenie k aplikácii výsledkov práce ako digitálne dvojča k reálnemu systému.

Kľúčové slova: Rubikova kocka, YuMi, ABB, RobotStudio, RAPID, Smart Component, digitálne dvojča

ABSTRACT

Abstrakt ve světovém jazyce

This bachelor thesis deals with the solution of the Rubik's Cube with the help of a collaborative dual-arm robot from the company ABB, under the name IRB 14000 YuMi. The work aims to achieve the solution of the Rubik's cube using a robot utilizing the same algorithm that would be used by a human, in contrast to the use of a computer-generated algorithm. The work deals with the history of the Rubik's Cube, existing solving algorithms and provides a deeper explanation of the used algorithm, but also its implementation in the RAPID programming language, the use of RobotStudio environment tools for solution visualization and last but not least the connection of results for a digital twin of the real system.

Keywords: Rubik's cube, YuMi, ABB, RobotStudio, RAPID, Smart Component, Digital Twin

Pod'akovanie

Týmto spôsobom by som chcel veľmi poďakovať hlavne vedúcemu bakalárskej práce Ing. Ľubošovi Spačkovi, za jeho trpezlivosť, vecné pripomienky a odborné rady pri vypracovaní práce. Taktiež veľká vďaka patrí mojej priateľke, rodine a kamarátom za podporu nielen počas písania práce, ale aj počas celého štúdia.

Motto

*„Decide what you want, decide what you are willing to exchange for it.
Establish your priorities and go to work.“*

-- H. L. Hunt --

OBSAH

ÚVOD	9
I TEORETICKÁ ČASŤ	10
1 RUBIKOVA KOCKA	11
1.1 ENRÓ RUBIK.....	11
1.1.1 Osobný život	11
1.1.2 Kariéra.....	12
1.2 HISTÓRIA KOCKY	12
1.2.1 Úspechy a ocenenia Rubikovej kocky	13
1.3 VÝVOJ KOCKY.....	15
1.4 POPIS KLASICKEJ RUBIKOVEJ KOCKY	17
1.5 DRUHY KOCIEK	22
1.6 PERMUTÁCIE A KOMBINAČNÉ CHARAKTERISTIKY	23
2 METÓDY SKLADANIA KOCKY	25
2.1 LEGENDA ROTÁCIÍ.....	26
2.2 ZAČIATOČNÍCKA METÓDA (BEGINNER'S METHOD).....	27
2.3 POKROČILÁ METÓDA (ADVANCED METHOD - CFOP).....	29
2.4 METÓDA ROUX (ROUX METHOD).....	30
2.5 METÓDA ZZ (ZZ METHOD)	31
2.6 METODA THISTLETHWAITE 45	31
2.7 GOD'S NUMBER	32
2.7.1 The Super-Flip	33
3 ROBOTIKA V PRIEMYSLE	34
3.1 ABB.....	34
3.1.1 O spoločnosti ABB	34
3.1.2 IRB 14000 YuMi.....	35
3.1.2.1 Technické parametre.....	35
3.1.2.2 SmartGripper	36
3.1.2.3 Spôsob komunikácie.....	37
3.1.3 RobotStudio.....	37
3.1.3.1 Smart Componenty	37
3.1.4 RAPID.....	39
3.2 DIGITÁLNE DVOJČA	39
II PRAKTICKÁ ČASŤ	40
4 IMPLEMENTOVANIE ALGORITMU DO KÓDU	41
4.1 ZÁPIS KOCKY.....	41
4.2 ROTÁCIE KOCKY.....	43
4.3 RIEŠENIE KRÍŽA NA SPODNEJ STRANE KOCKY	44
4.3.1 Prípád umiestnenia hľadanej hrany dole, orientovanej správne na správnej alebo nesprávnej pozícií	45
4.3.2 Prípád umiestnenia hľadanej hrany hore, orientovanej správne	46
4.3.3 Prípád umiestnenia hľadanej hrany v jednej zo stien.....	47
4.3.4 Prípád umiestnenia hľadanej hrany dole	49

4.3.5	Prípád umiestnenia hľadanej hrany hore.....	51
4.4	RIEŠENIE PRVEJ VRSTVY	52
4.5	RIEŠENIE DRUHEJ VRSTVY	54
4.6	ORIENTÁCIA POSLEDNEJ VRSTVY	56
4.7	PERMUTÁCIE ROHOV POSLEDNEJ VRSTVY	59
5	IMPLEMENTÁCIA LOGIKY	60
5.1	ROBTARGETY	60
5.2	SYNCHRONIZÁCIA	60
5.3	APLIKOVANIE ZÍSKANÉHO RIEŠENIA.....	61
6	SMART COMPONENT KOCKY.....	63
6.1	REALIZÁCIA	63
7	SIMULÁCIA A VÝSLEDKY.....	67
7.1	OVLÁDANIE.....	67
7.2	PRIEMERNÝ POČET POHYBOV	68
7.3	PRÍPADNÉ VYLEPŠENIA.....	68
ZÁVER		69
ZOZNAM POUŽITEJ LITERATÚRY		70
ZOZNAM POUŽITÝCH SYMBOLOV A SKRATIEK.....		74
ZOZNAM OBRÁZKOV		75
ZOZNAM PRÍLOH.....		77

ÚVOD

Roboty sa stali súčasťou nášho každodenného života. Nielen pre svoju všestrannosť a adaptívnosť, ale aj pre presnosť vykonávania požadovaných inštrukcií a nepretržitej prevádzky, sa stali obrovským lákadlom pre mnoho firmy a priemysel. Dôležitú úlohu zohrávajú napríklad v medicíne, kde sú nasadené pri zákrokoch, ktoré by pre bežného lekára boli príliš zložité. Obrovské zastúpenie majú už v spomínanom priemysle, kde nahrádzajú ľudí pri zdravotne nebezpečných a fyzicky náročných úkonoch. V poľnohospodárstve pri vykonávaní stereotypných prác, spojených s udržiavaním a pestovaním úrody. Svoje využite našli aj v bežnom živote pri rozmanitých domácich prácach.

Čoraz viac sa používajú kolaboratívne roboty, ktoré aj napriek obmedzeniam v nosnosti a dosahu, kvôli splneniu požadovanej bezpečnosti, sú schopné spolupracovať s ľuďmi pri výrobe ale aj rozličných činnostiach. Spolupráca s ľuďmi však nemusí byť podmienkou ich využitia. V tejto práci je použitý kolaboratívny dvojramenný robot od firmy ABB s označením IRB 14000 YuMi, ktorý po zamiešaní Rubikovej kocky, s využitím naprogramovaného algoritmu, je schopný kocku vyriešiť, v podobnosti s riešením ľudským. Celé riešenie je navrhnuté v simulačnom prostredí.

Teoretická časť práce v kapitole 1 približuje život vynálezcu Rubikovej kocky, históriu, vývoj a popis kocky ale aj matematickú charakteristiku. Kapitola 2 je zameraná na typy vybraných metód skladania kocky. Kapitola 3 popisuje spoločnosť ABB a použitého robota spolu s použitým uchopovačom. Taktiež ukazuje funkcie programu RobotStudio a programovacieho jazyka RAPID.

Praktická časť práce začína kapitolou 4, ktorá sa zaoberá implementáciou algoritmu do kódu. Vysvetlený je výber použitých metód skladania kocky a reprezentácia kocky. Kapitola 5 nadväzuje na predchádzajúcu tým, že rieši aplikovanie získaného riešenia a taktiež pohyby robota v simulácii. V kapitole 6 je vysvetlené fungovanie simulovaného modelu kocky. Praktickú časť uzatvára kapitola 7, v ktorej je ukázané ako je možné spustiť celkovú simuláciu a sú v nej poukázané na prípadné vylepšenia.

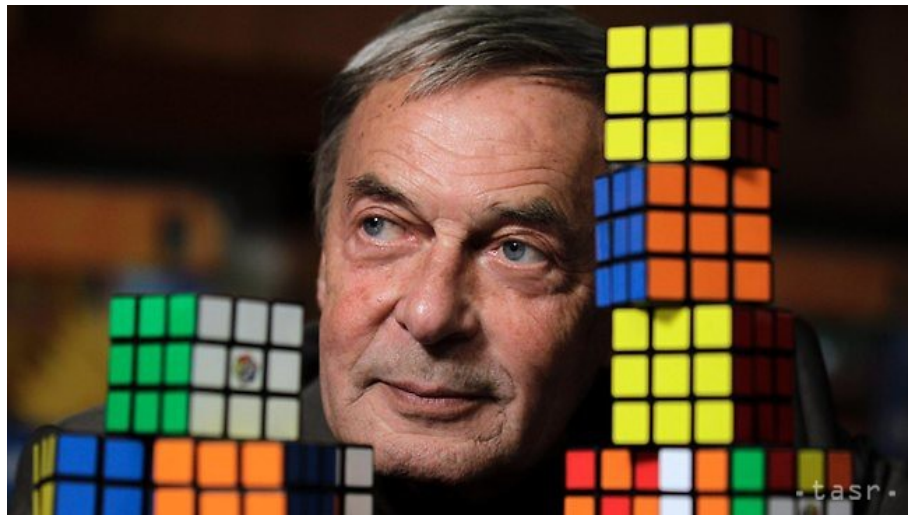
I. TEORETICKÁ ČASŤ

1 RUBIKOVA KOCKA

Rubikova kocka od svojho zrodu k dnešnej podobe prešla rôznymi štádiami vývoja, o ktorých sa možno dočítať v tejto kapitole.

1.1 Ernő Rubik

Vynálezca Rubikovej kocky, Ernő Rubik (*Obr. 1*) sa narodil 13. júla 1944 v hlavnom meste Maďarska, Budapešti. Deň 13. júl sa na jeho počesť stal dňom hlavolamov a rébusov. Tento maďarský architekt a dizajnér má pôvod na Slovensku, konkrétne v meste Piešťany. [1]



Obr. 1: Ernő Rubik [1]

1.1.1 Osobný život

Z Piešťan pochádza jeho otec Ernő Rubik st., ktorý bol letcom a leteckým konštruktérom. Pomáhal pri vytváraní leteckého priemyslu v Maďarsku po druhej svetovej vojne. Taktiež sám navrhol niekoľko vetroňov pre leteckú asociáciu Műegyetemi Sportrepülő Egyesület, ktorá spadá pod Budapeštiansku technickú univerzitu. Model R-26 Góbé (*Obr. 2*) je jeden z jeho najznámejších vetroňov. Vďaka svojej produktivite sa neskôr stal jedným z najznámejších leteckých konštruktérov v Maďarsku. Po istom čase si otvoril vlastný podnik, kde svoje lietadlá vyrábal. V roku 1963 sa mu podarilo získať Kossuthovu cenu, čo je najvyššie maďarské ocenenie pre vedu. Matka Ernóa Rubika, Magdolna Szánto, bola spisovateľkou a s veľkou obľubou sa venovala poézii. [1][2]



Obr. 2: Vetroň R-26 Góbé [2]

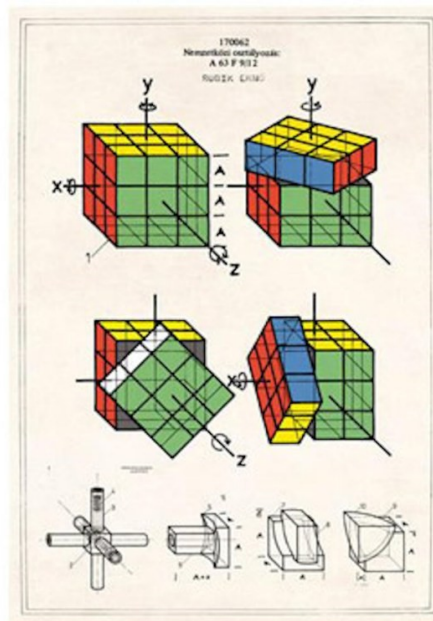
1.1.2 Kariéra

Rubik študoval sochárstvo na Technickej univerzite v Budapešti a architektúru na Akadémii aplikovaných umení a designu, tiež v Budapešti. Bol šikovným študentom, ktorého zaujímala technika, geometria a vesmír. Po ukončení štúdií sa stal profesorom designu na akadémii. Svojich študentov nabádal chápať geometriu priestorovo, na čo používal trojrozmerné modely vyhotovené z papiera, dreva alebo lepenky – tak poskytol študentom vizualizácie. Postupne viac prepadal svojmu hobby, stavaniu geometrických modelov. Jeden z týchto modelov bol prototypom “jeho“ kocky, ktorý bol vytvorený z 27 drevených blokov. K dosiahnutiu dnešnej podoby Rubikovej kocky predchádzalo viacero prototypov a rôznych verzií. [1][3]

1.2 História kocky

Celý zrod tejto kocky prebehol v roku 1974, keď Rubik vyučoval na umeleckopriemyselnej vysokej škole a snažil sa vynájsť niečo, aby mohol študentom na príklade podrobnejšie vysvetliť trojrozmerný priestor. V tejto dobe prototyp kocky ešte neniesol svoj svetoznámy názov Rubikova kocka, ale bol označovaný ako Magická kocka. Od dnešnej podoby kocky sa tento prototyp nijako dramaticky nelíšil. Mal síce skosené hrany kvôli o niečo väčším rozmerom a bol vyrobený z dreva, ale zložením pripomínal dnešný originál. Pridaním 54 farebných nálepiek bol vytvorený ikonický výzor. Po prvom zamiešaní kocky Rubikovi trvalo vyše mesiaca, aby ju vrátil do pôvodného stavu. V roku 1975 si Rubik kocku nechal patentovať, stále pod pôvodným názvom (*Obr. 3*). Kocka sa po prvýkrát začína predávať v Budapeštianskom hračkárstve v roku 1977 a už dosahuje aj určitej kvality. Kvalita, ktorá zabezpečovala, aby sa kocka počas otáčania nerozpadla a bola vhodná aj pre deti. V roku

1979 bola kocka predstavená na mnohých medzinárodných veľtrhoch, zahrňujúc veľtrh v Londýne, Paríži a New Yorku. V septembri toho istého roku si kocku všimol, a videl v nej veľký potenciál, aj špecialista na hračky Tom Kremer v Norimbergu. Trvalo niekoľko rokov, kým distribúcia tohto vynálezu prerazila do sveta. Dôvodom spomínaného pozastavenia bola Železná opona, ktorá v tom čase oddeľovala Európu pred úpadkom komunizmu. Nakoniec sa podarilo v roku 1980 kocke svetovo preraziť, keď bola podpísaná medzinárodná dohoda medzi hračkárstvami po celom svete. Aj vďaka vášni a viere Toma Kremera, ktorý presvedčil spoločnosť Ideal Toy Company aby kocku distribuovala, sa tento sen stal skutočnosťou. Avšak spoločnosť chcela nový, autentickjší názov. Magická kocka oficiálne získava meno Rubikova kocka. Vďaka svojim jedinečným vlastnostiam táto kocka spôsobila veľký celosvetový záujem, ktorý ľudstvo predtým ešte nezažilo. V tom istom roku vzniká aj prvý algoritmus na vyriešenie kocky vytvorený Davidom Singmasterom, nazývaný “vrstva po vrstve“, ktorý sa používa dodnes. [4][5][6]



Obr. 3: Patent Ernőa Rubika –
Magická kocka ako puzzle [4]

1.2.1 Úspechy a ocenenia Rubikovej kocky

V rokoch 1980 a 1981 bola kocka ocenená špeciálnou cenou Nemecká hra roka a taktiež získala podobné ocenenia v Spojenom kráľovstve, Francúzsku a v Spojených štátoch amerických. Aj vďaka týmto oceneniam sa Rubikova kocka považuje za jednu z najlepšie predávaných hračiek na svete. [4][6]

Do dnešného dňa bolo predaných vyše 350 miliónov kusov. Zároveň je v roku 1981 publikovaná kniha s názvom *You Can Do The Cube* od 12 ročného autora Patricka Bosserta, ktorý ich predal vyše 1,5 milióna kópií. V roku 1982 sa uskutočnila prvá celosvetová súťaž v Budapešti, kde sa jednotlivci pokúsili o zloženie kocky v čo najkratšom čase. Víťazom súťaže sa stal americký tínedžer Minh Thai, ktorému sa podarilo kocku zložiť za 22,95 sekúnd. Sám po tomto úspechu vydal knihu s názvom *The Winning solution*, ktorá neskôr slúžila aj ako podklad pre metódu *The Ortega (Corners-First)*. [4][6]

V nasledujúcich rokoch sa Rubikova kocka dostáva do popredia záujmu. Vyskytuje sa v rôznych seriáloch a filmoch, medzi ktoré patria „*The Simpsons*“, „*Dude, Where’s My Car?*“, „*Wall-E*“ ale aj film „*The Pursuit of Happiness*“, v ktorom sa s kockou hrá Will Smith. Za zmienku stojí aj výstava pouličného umelca z roku 2005 „*Space Invader*“ (*Obr. 4*) na tému „*RubikCubism*“, kde boli vystavené rôzne obrazy vyskladané z Rubikových kociek. V týchto rokoch zároveň kocka zažíva aj úpadok v dôsledku výroby nových, o niečo modernejších hračiek. Rubik neskôr vydal aj iné hračky, viac známy „*The Snake*“ v roku 1983 alebo menej známy hlavolam „*Magic*“ v roku 1987 (*Obr. 5*). [4]



Obr. 4: *Space Invader* – Výstava pouličného umelca [4]



Obr. 5: The Snake a hlavolam Magic [4]

Medzičasom sa uskutočnilo mnoho súťaží, na ktorých padli rekordné časy v skladaní koc-
ky. Forma tohto súťaženia sa anglicky nazýva „Speedcubing“. Víťazom turnaja organizo-
vaného v Bangkoku v roku 2011 sa stal poľský súťažiaci Michal Pleskowicz s časom 8,65
sekundy, keď porazil vtedajší svetový rekord. Tento rekord mu prebral americký tínedžer
Lucas Etter s časom 4,9 sekundy v roku 2015. Za tri roky padol ďalší rekord, ktorý patril
Austráľčanovi Feliksovi Zemdegessovi s časom 4,22 sekundy. Doposiaľ za najrýchlejší čas
3,47 sekundy sa v roku 2018 podarilo poskladať kocku Číňanovi Yusheng Duovi. Yusheng
Du drží svetový rekord až do súčasnosti. [4]

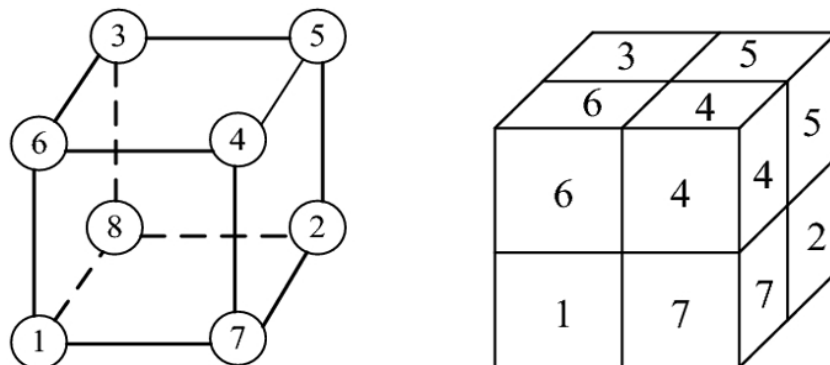
1.3 Vývoj kocky

Pôvodne koncept Rubikovej kocky čerpá z čínskej kultúry, konkrétne z „China Luo Book“,
ktorý môže byť zjednodušený do Jiugong mapy. Jiugong mapa (Obr. 6) predstavuje nula-
rozmerovú, kocku tretieho radu. [7]

4	9	2
3	5	7
8	1	6

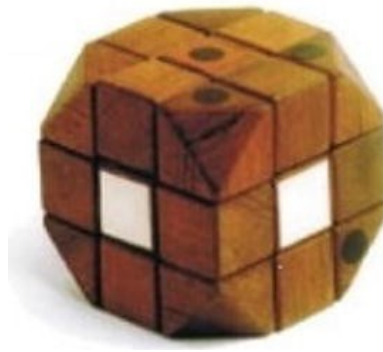
Obr. 6: Jiugong mapa [7]

Magický štvorec n -tého radu je určitá konfigurácia vytvorená kombináciou čísel $1, 2, 3, \dots, n^2$ v štvorci n -tého radu, čo vytvára súhrn čísel v každom riadku, stĺpci, a v dvoch diagonálach: $n(n^2+1)/2$. Tento postup je nazývaný magická štvorcová konštanta. Magická štvorcová konštanta kocky tretieho radu má hodnotu 15. „Rearrange Jiugong“ predstavuje jednorozmernú kocku tretieho radu. Je to hra založená na základoch Jiugong mapy pravdepodobne z obdobia čínskej dynastie Yuan. Táto hra ukladá osem pohyblivých kúskov/figúrok na deväť miest v hracej ploche tak, aby sa premiešal pôvodný vzor. Princípom hry je posúvanie figúrok, dokým sa vzor nevráti do pôvodného stavu. Spomínaná hra sa časom rozšírila na západ do Európy, kde vďaka nej Sam Loyd vynášiel „15 chess“. Princíp je ten istý - posúvať figúrky za cieľom vzorovej zmeny. Rozdiel je len v tom, že „15 chess“ je zostavený štvorradovo, čiže má o jednu radu navyše. Neskôr sa vývoj kocky presunul z radového poňatia na rozmerové. V dynastii Čching čínski vedci navrhli myšlienku použitia trojrozmerného magického štvorca, čo viedlo k vytvoreniu trojrozmernej kocky druhého radu (Obr. 7). [7]



Obr. 7: Trojrozmerný magický štvorec a kocka druhého radu [7]

Do roku 1974 nemal trojrozmerný magický štvorec rotačnú charakteristiku, pokým Rubik nenavrhol rotačný mechanizmus kocky. Rovnaký dizajn dokončil v roku 1976 aj japonský Stonefur. Na dosiahnutie vzorových zmien sa využíva princíp pohybu jednotlivých dielov, podobný ako pri hre „Rearrange Jiugong“. Základom tohto princípu je dráha, po ktorej sa diely pohybujú a ich pohyb je obmedzený inými dielmi, ktoré sú s nim v kontakte. Rubik vytvoril prvú trojrozmernú kocku tretieho radu na svete. Výzorom pripomínala guľu kvôli skoseniu ôsmich vrcholov kocky, ktoré boli vytvorené z bezpečnostného hľadiska (Obr. 8). Neskôr sa tvar kocky zmenil – oblé rohy sa stali ostrými a tak vznikol kubický tvar. Táto zmena nijako neovplyvnila funkcionálnosť kocky a jej výrobný proces sa stal jednoduchším. [7]

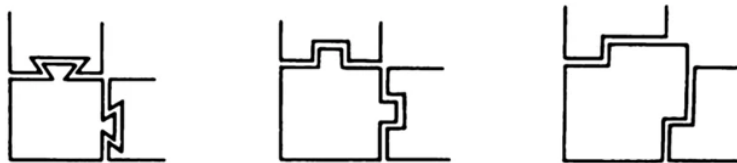


Obr. 8: Stará verzia Rubikovej kocky [7]

Druhý prototyp kocky, ktorej rotácia sa spoliehala na magnetizmus, bol vynájdený už v roku 1959. Tento druh kocky je však menej populárny ako Rubikova kocka. Na jednej strane z dôvodu magnetickej štruktúry sú náklady na výrobu tejto kocky príliš vysoké. Na druhej strane druhý prototyp kocky sa veľmi ťažko opravuje. O nejaký čas Rubik vynachádza tretí prototyp kocky, ktorý má dômyselný dizajn, kompaktnú štruktúru a nízke náklady na výrobu. Tým prekonáva nedostatky magnetickej kocky druhého radu. Stručne povedané, k vynaliezavosti Rubikovej kocky je kľúčom jej špeciálna mechanická štruktúra. [7]

1.4 Popis klasickej Rubikovej kocky

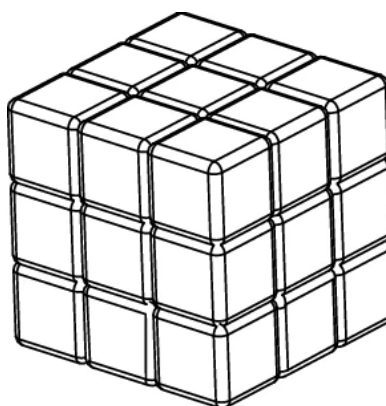
Rubikova kocka je svetovo populárne mechanické puzzle, ktoré si získalo svoju pozornosť vďaka svojim unikátnym charakteristikám. Okrem klasickej hračky na tréning mozgu je tento predmet využívaný aj na vedecké výskumy a technický vývoj. Pod pojmom Rubikova kocka si každý z nás predstaví kocku 3x3x3. Tá pozostáva z 26 menších kociek a na 27 pozícií sa nachádza stredové držadlo kocky/stredový hriadeľ. Kocky sa líšia šiestimi rôznymi farbami. Konkrétne prvá Rubikova kocka niesla tieto farby: červenú, modrú, oranžovú, zelenú, žltú a bielu. Najkrajším aspektom Rubikovej kocky je štruktúra jej mechanizmu. Len malé množstvo ľudí navrhlo možný mechanizmus a ešte menej návrhov bolo aj praktických. Neskôr bolo mnoho mechanizmov založených na spojoch medzi čapom a drážkou. Na obrázku (*Obr. 9*) môžeme vidieť spojenia Rubikovej kocky. Čapovacie a dlabacie metódy sa dajú použiť tak, že 26 vonkajších častí kocky je držaných pohromade bez stredového kusu. Tento spoj je náročné vytvoriť s dostatočnou presnosťou tak, aby sa steny ľahko otáčali. [7]



a Dovetail tenon **b** Square tenon **c** Shouldered tenon

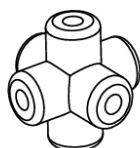
Obr. 9: Tri druhy drážky a čapu – spojenia Rubikovej kocky [7]

Klasická kocka 3x3x3 (Obr. 10), ako už bolo poznamenané, pozostáva z 26 jedinečných menších kociek, vrátane rámu stredového hriadeľa, 6 stredových dielov, 8 rohov a 12 hrán. [7]

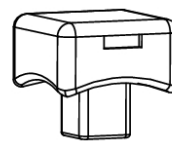


Obr. 10: Rubikova kocka [7]

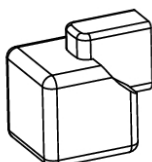
Rôzne typy dielov Rubikovej kocky sú vyobrazené na obrázku (Obr. 11). Každý tento diel obsahuje skryté, dovnútra predĺžené spojenie. To sa blokuje s ostatnými dielmi, a zároveň im umožňuje pohyb na rôzne miesta. [7]



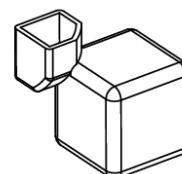
a Center shaft frame



b Center piece



c Edge piece

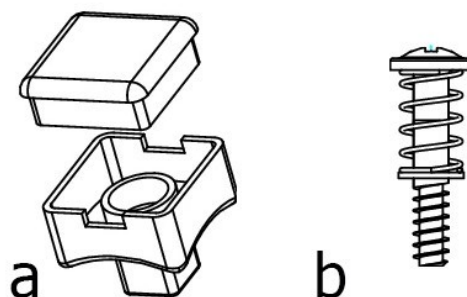


d Corner piece

Obr. 11: Diely Rubikovej kocky [7]

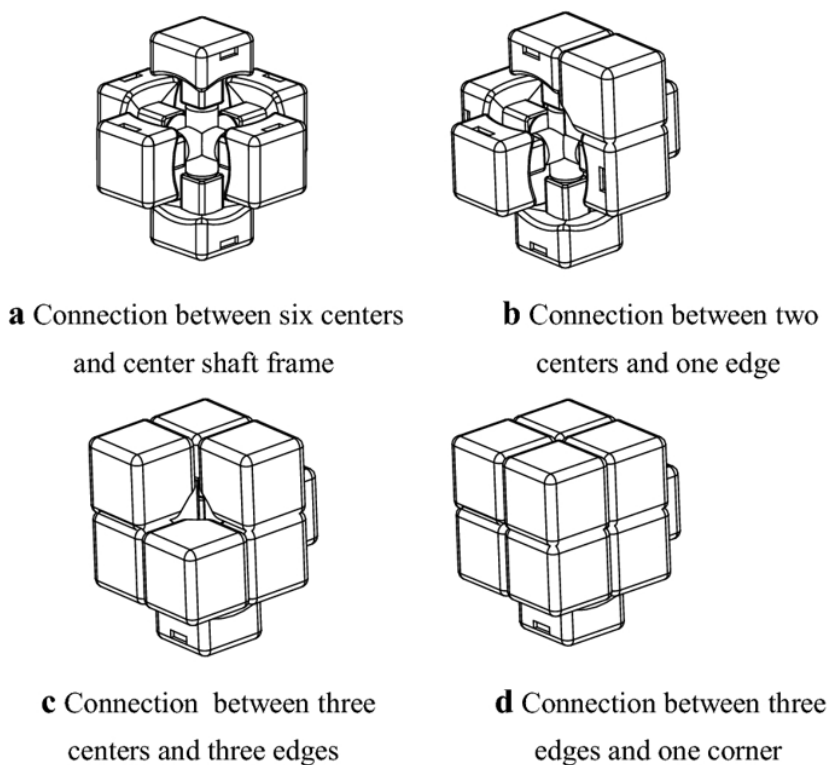
Medzi diely Rubikovej kocky z obrázka (Obr. 11) patria: a – rám stredového hriadeľa, b – stredový diel, c – hranový diel, d – rohový diel.

Obrázok (Obr. 12 a) znázorňuje, ako môže byť stredový diel rozdelený na dve časti. Stredový diel a rám stredového hriadeľa sú spojené pružinovými skrutkami, ktoré sú znázornené na obrázku (Obr. 12 b). [7]



Obr. 12: Časti centrálného dielu [7]

Obrázok (Obr. 13) nám predstavuje postup zostavovania rôznych dielov. Obmedzenia po okrajoch sú tvorené štrukturálnymi obmedzeniami a silovým blokovaním dvoch susedných stredov. Obmedzenia rohového dielu sú tvorené štrukturálnymi obmedzeniami a silovým blokovaním troch susedných hrán. [7]

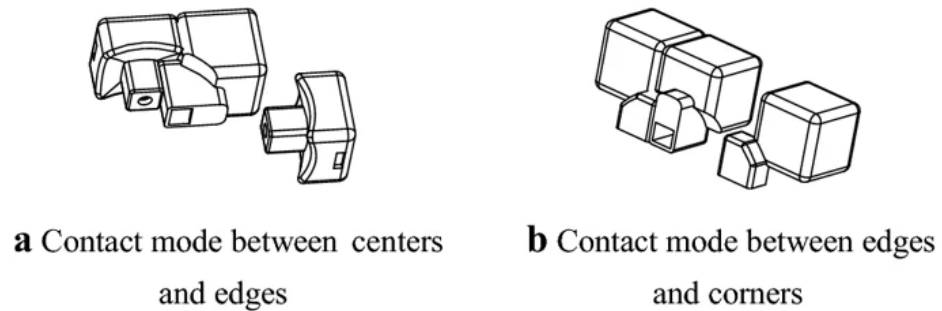


Obr. 13: Spojenia v Rubikovej kocke [7]

Medzi spojenia v Rubikovej kocke z obrázka (Obr. 13) patria: a – spojenie medzi šiestimi stredmi a rámom stredového hriadeľa, b – spojenie medzi dvoma stredmi a jednou hranou,

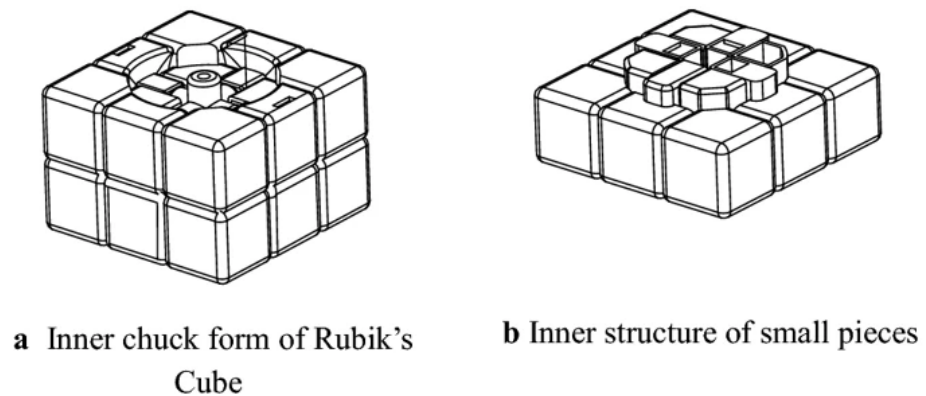
c - spojenie medzi troma stredami a troma hranami, d – spojenie medzi troma hranami a jedným rohom.

Režimy kontaktu medzi stredmi a hranami sú znázornené na obrázku (Obr. 14 a), režimy kontaktu medzi rohmi a hranami sú znázornené na obrázku (Obr. 14 b). Všetky sa navzájom dotýkajú zakriveným povrchom a rovinou. [7]



Obr. 14: Režimy kontaktu [7]

Vnútorne časti rohov a hrán sú pevne spojené. Tvoria spolu vnútorné skľučovadlo, ktoré môžeme vidieť na obrázku (Obr. 15 a). Obrázok (Obr. 15 b) znázorňuje deväť malých dielov na jednej strane, ktoré sú pevne zovreté tak, že sa od seba nerozchádzali. Na druhej strane sa vrstva Rubikovej kocky pôsobením sily môže otáčať okolo stredového hriadeľa. [7]

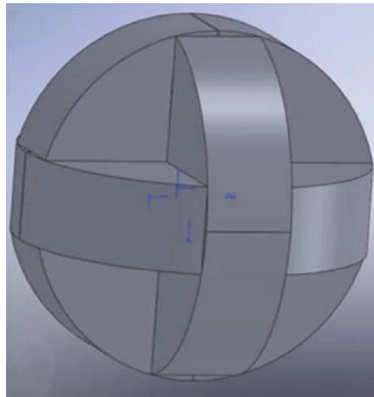


Obr. 15: Vnútorne časti Rubikovej kocky [7]

Zaujímavosťou je, že charakteristika Rubikovej kocky, ktorej malé diely sú spojené spoločnou mozaikou, je podobná Lubanskému zámku. Lubanský zámok (Obr. 16) je tradičná čínska inteligentná hračka, ktorá pochádza z kostrových štruktúr čínskej starodávnej architektúry. Časti tejto hračky sú štrukturálnym obmedzením držané pohromade, a to bez akýchkoľvek klincov alebo šnúrok. Avšak vzťah medzi rôznymi časťami je komplikovaný. Nasledujúci obrázok predstavuje guľový Lubanský zámok. Je zostavený zo šiestich dreve-

ných částí (*Obr. 17*), které mohou být spojené dohromady. V tomto případě jde o zostavu, kde sú drevené časti navzájom spojené. [7]

Zároveň ostáva priestor pre zasunutie zadných komponentov až do umiestnenia poslednej časti „kľúčovej dierky“. Celý zámok je pevne držaný pohromade – šesť častí sa navzájom podopiera a zadržiava. Od drevenej štruktúry Lubanského zámku je odvodená kombinovaná štruktúra Rubikovej kocky. [7]

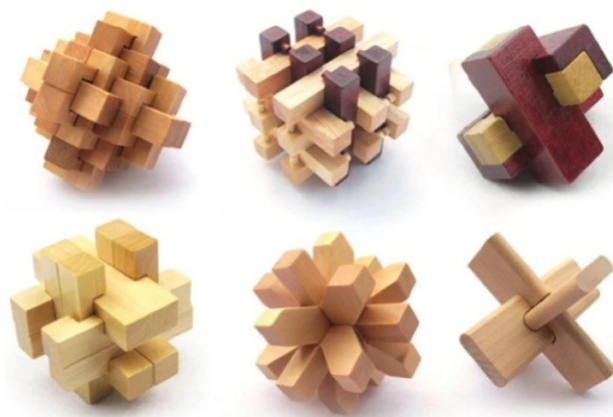


Obr. 16: Lubanský zámok
po zložení – lopta [7]



Obr. 17: Časti Lubanského zámku – lopta [7]

Šesť druhov Lubanskej zámkovej štruktúry je znázornených na obrázku (*Obr. 18*). Štruktúry sú spojené metódou spájania konkávne-konvexného spracovania. To dokazuje veľkú kreativitu a úspechy staroveku. [7]



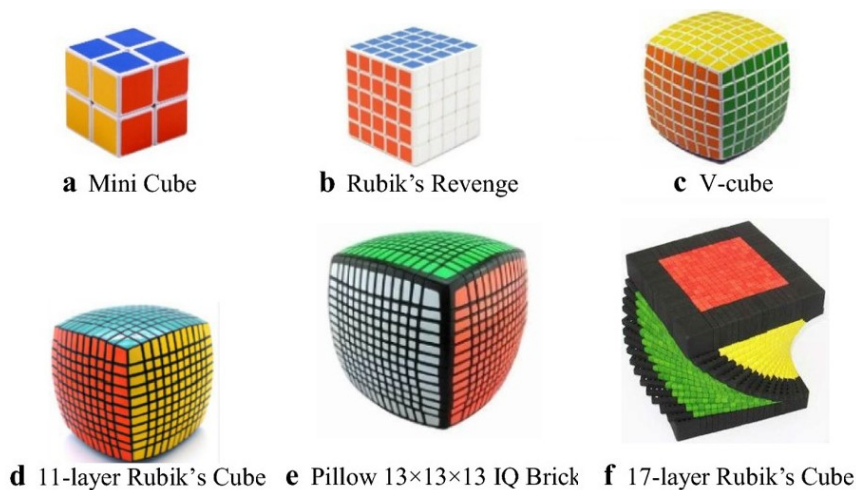
Obr. 18: Lubanské zámky [7]

Stabilita dlabu a štruktúra čapu je ovplyvnená faktormi, medzi ktoré patrí napríklad umiestnenie drážok a tesnosť jednotlivých častí. Kvalita spojenia maltovej a tenónovej štruktúry závisí od toho, ako sú kúsky rôznej veľkosti presne umiestnené dohromady. Ak sa v týchto prípadoch odstráni drevená tyč, celý zámok sa rozpadne. Na dosiahnutie celkovej stability používa Lubanský zámok aj Rubikova kocka mozaikové štruktúry. Rozdiel medzi týmito dvoma prípadmi spočíva v tom, že stena Rubikovej kocky môže voľne rotovať, zatiaľ čo Lubanský zámok to nedokáže. [7]

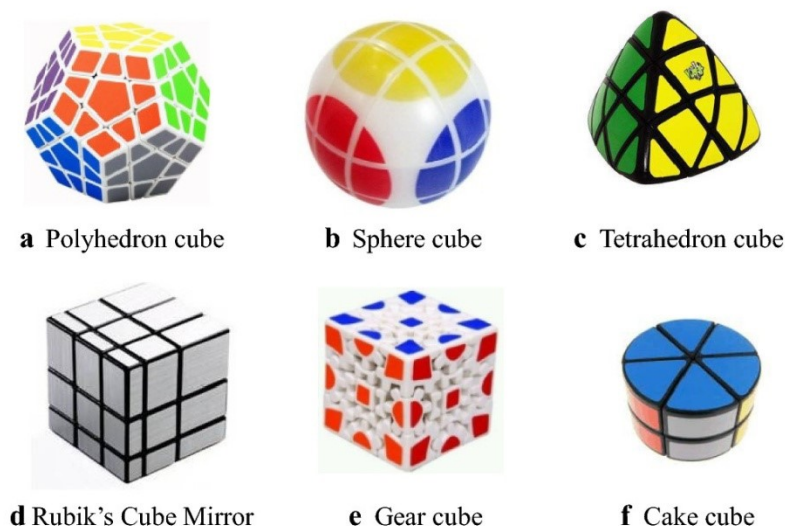
1.5 Druhy kociek

Množstvo rôzne upravených kociek rozširujúcich túto kockovú rodinu bolo navrhnutých po vynájdení Rubikovej kocky. Vo všeobecnosti môžeme tieto kocky rozdeliť do dvoch kategórií - kocky kubické a špeciálne tvarované. Kubická kocka má typický pravidelný tvar, ktorý sa nemení, ale počet vrstiev sa meniť môže. V dnešnej dobe existujú rôzne variácie Rubikových kociek, ktoré môžu mať až 33 vrstiev. Vieme ich zoradiť vzostupne, podľa počtu usporiadania vrstiev. Prvá, najmenšia kocka má počet kostičiek $2 \times 2 \times 2$. Je známa ako vrecková alebo mini kocka. Ďalšou v poradí, tou štandardnou, je kocka $3 \times 3 \times 3$. Nasleduje kocka $4 \times 4 \times 4$, tiež známa ako „Rubik’s Revenge“ alebo „Master Cube“. Kocka $5 \times 5 \times 5$ sa nazýva „Professor’s Cube“. Tieto kocky sú najznámejšie a sú predajne dostupné na mnohých internetových obchodov za prijateľné ceny. [7] Medzi ďalšie, známejšie viacvrstvové kubické kocky radíme $17 \times 17 \times 17$, ktorá sa nazýva „Over the Top“. Táto kocka bola v roku 2011 vytvorená Oskarom van Deventerom a taktiež bola zapísaná do Guinnessovej knihy rekordov. Jej cena sa pohybovala okolo 600 dolárov. [8] O päť rokov neskôr bol tento rekord pokorený, vytvorením kocky $22 \times 22 \times 22$. [7] Ani nie dva roky na to, v decembri 2017, bola predstavená kocka $33 \times 33 \times 33$, ktorá si do dnešných dní drží svoje prvenstvo. Túto kocku vytvoril francúzsky mechanický inžinier Grégoire Pfennig, ktorý jej jednotlivé časti vytlačil na 3D tlačiarňi. Vytlačených častí dokopy bolo 6 153 kusov, použitých bolo 6 534 nálepiek a celkové zhotovenie kocky trvalo Pfennigovi 205 hodín. [9] Ďalšiu kategóriu tvoria špeciálne tvarované kocky s dodatočným vybavením, ako napríklad s bluetooth sledovaním. Osobitú skupinu tvoria špeciálne kocky kockovej rodiny, okrem kubickej. Tieto kocky majú špeciálne štruktúrne tvary, ktoré sú rozmanité, vrátane mnohostenných hlavolamov, guľového hlavolamu, zrkadlovej kocky, ozubenej kocky, koláča a podobne. [7]

Obrázok (Obr. 19) zobrazuje kubické druhy Rubikovej kocky a obrázok (Obr. 20) zobrazuje špeciálne druhy Rubikovej kocky.



Obr. 19: Druhy Rubikovej kocky – kubické [7]



Obr. 20: Druhy Rubikovej kocky – špeciálne [7]

1.6 Permutácie a kombinačné charakteristiky

Rohy môžu byť usporiadané 40 320 (8!) rôznymi spôsobmi s 3^7 možnými orientáciami, pretože orientácia ôsmeho – posledného rohu závisí od predchádzajúcich kombinácií rozložení rohov. Hrany môžu byť usporiadané do 239 500 800 (12!/2) spôsobov. Obmedzenie na 12! je preto, lebo hrany musia byť v rovnomernej permutácii presne vtedy, kedy aj rohy. Existuje 2048 (2^{11}) možností preklopenia hrán. 11 hrán môže byť preklopených nezávisle s preklopením dvanástej hrany. Postupnými rotáciami jednotlivých strán sa získajú rôzne

kombinácie usporiadaní daných kociek. Počet rôznych kombinácií zamiešania Rubikovej kocky je teda

$8! \times 3^7 \times (12!/2) \times 2^{11} = 43\,252\,003\,274\,489\,856\,000$, čo je okolo 43 triliónov. [5][10]

Existuje 43 252 003 274 489 856 000 rôznych kombinácií zamiešanej kocky. Keby každý človek na planéte, každú sekundu vytvoril jednu kombináciu, tak by trvalo 176 rokov vytvoriť všetky kombinácie Rubikovej kocky.

Pre lepšiu predstavu sa dá ľahko spočítať, že keby niekto mal jednu štandardnú veľkosť Rubikovej kocky pre každú permutáciu, mohol by tento človek pokryť zemský povrch 275-krát, alebo naskladať vežu vysokú 261 svetelných rokov. Predchádzajúca hodnota je obmedzená na permutácie, ktoré môžu byť dosiahnuté iba otočením strán kocky. Ak vezmeme v úvahu permutácie, ktoré by sme dosiahli demontážou kocky, ich počet sa zvýši dvanásťkrát, čo je približne 519 triliónov. Ako režim, ktorý využíva myšlienku permutácie a kombinácie na sledovanie rozmanitostí odpovedí, sa môže použiť transformácia medzi rôznymi konfiguráciami Rubikovej kocky. Vlastnosti transformácie a konfigurácie inšpirujú myslenie pri návrhu produktu. Zároveň majú praktickú aplikačnú hodnotu a to najmä pri modulácií priemyselného produktu a pri konštrukčnom navrhovaní. Jednou zo základných charakteristík Rubikovej kocky je cirkulácia. Ak sa použije ľubovoľná postupnosť operácií (rotačná sekvencia) na cyklické ovládanie kocky, dosiahne sa stav cyklu. Táto cyklická charakteristika sa dá rozdeliť do dvoch kategórií: periodickej a neperiodickej. Za predpokladu, že kocka je riadená určitými číselnými operáciami a špecifickými operáciami z pôvodného stavu späť na pôvodný stav, znamená to, že kocka dosahuje cyklus. Počet otáčok Rubikovej kocky je jedno obehové obdobie kocky. Ak je doba obehu konštantná, cyklus je periodický. Ak je doba obehu variabilná, cyklus je neperiodický. Sekvencia operácií neperiodického cyklu je založená na postupnosti periodických cyklických operácií a na postupnosti operácií „konštantného procesu“, ktoré spôsobujú, že vzor kocky vykazuje neperiodickú zmenu. Postupnosť operácie „konštantný proces“ sa vzťahuje na skutočnosť, že všetky kusy sa môžu po tejto postupnosti operácií vrátiť do pôvodného stavu. [5]

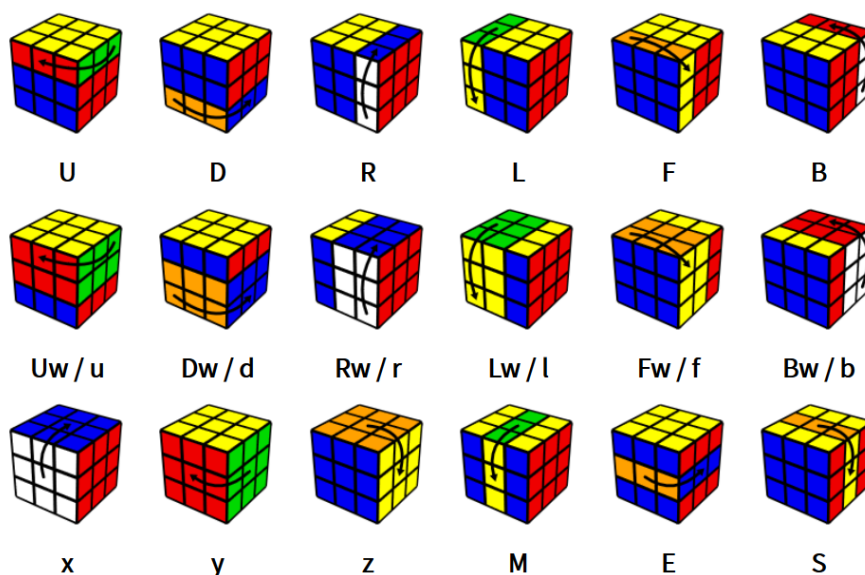
2 METÓDY SKLADANIA KOCKY

Už od svojho vzniku je Rubikova kocka známa svojím jedinečným šarmom ako klasická vzdelávacia hračka. [7] Od všeobecne známych postupov riešenia, cez čo najkratší čas a najmenší počet pohybov, po riešenie kocky jednou rukou alebo riešenie so zaviazanými očami, stále viac a viac nadšencov Rubikovej kocky a súťažiacich hráčov sa zaujíma ako riešiť čoraz väčšie výzvy. [11] Sekvencia pohybov, ktorá má požadovaný efekt na kocku, sa v Rubikovej kuberskej reči nazýva algoritmus. Táto terminológia je odvodená z matematického použitia algoritmu. Samotné slovo „algoritmus“ znamená zoznam dobre definovaných inštrukcií na vykonávanie úlohy z daného počiatočného stavu, cez dobre následne definované stavy do požadovaného konečného stavu. Každá metóda riešenia Rubikovej kocky využíva svoju vlastnú sadu algoritmov spolu s opismi toho, aký účinok má algoritmus a kedy sa dá použiť na priblíženie kocky k riešeniu. Mnoho algoritmov je určených na transformáciu iba malej časti kocky bez zasahovania do iných častí, ktoré už boli vyriešené. To znamená, že sa môžu opakovane aplikovať na rôzne časti kocky, až kým sa nevyrieši celok. Napríklad existujú dobre známe algoritmy na cyklovanie troch rohov bez zmeny zvyšku puzzle alebo preklopenia orientácie páru hrán, zatiaľ čo ostatné zostanú nedotknuté. Niektoré algoritmy majú na kocku určitý požadovaný účinok (napríklad výmena dvoch rohov), ale môže mať tiež vedľajší účinok pri zmene iných častí kocky (napríklad prepúšťanie niektorých hrán). Takéto algoritmy sú často jednoduchšie ako algoritmy bez vedľajších účinkov a používajú sa na začiatku riešenia, keď väčšina skladieb ešte nebola vyriešená a vedľajšie účinky nie sú dôležité. Väčšina z nich je dlhá a ťažko zapamätateľná. Ku koncu riešenia sa namiesto toho používajú špecifickejšie (a zvyčajne komplikovanejšie) algoritmy. [5] Zákon „prvej vrstvy“ vynašiel matematik David Singmaster a používa ho veľa ľudí. Tento zákon spočíva v takomto postupe: najprv sa vyrieši vrchná vrstva, potom stredná a nakoniec aj spodná vrstva. [12] Ďalšiu metódu riešenia navrhol matematik John H. Conway. Jeho riešenie spočíva v šesťstupňovej redukčnej metóde, od spodnej časti kocky do stredu a potom na vrchol, a potom od okrajových častí po rohové kusy postupne. [13] Patrick Bossert navrhol šesťstrannú redukčnú metódu: najprv sa vyriešia rohy a hrany hornej vrstvy, potom sa vyriešia spodné rohy a nakoniec sa vyriešia hrany strednej a dolnej časti. [14] Za veľmi účinnú metódu sa považuje aj postup Jessicy Friedrich, známy ako „Fridrichova metóda“. Používa ju veľa pretekárov na rýchle vyriešenie Rubikovej kocky. Táto metóda sa využíva v širokom zábere, ovplyvňuje a inšpiruje výskumníkov kocky a zároveň odohrala rozhodujúcu úlohu pri jej vývoji. [7] Algoritmus

Rubikovej kocky je možné vyriešiť aj pomocou počítačovej umelej inteligencie. Richard E. Korf použil techniku „makro operátora“ na vyriešenie algoritmu Rubikovej kocky počítačovým programom. [15]

2.1 Legenda rotácií

Pre značenie rotácií kocky sa používajú písmená abecedy, ktoré aj označujú jednotlivé steny. Nimi sú: **F** (front), **U** (up), **R** (right), **B** (back), **D** (down) a nakoniec **L** (left). Potom rotácie v smere hodinových ručičiek o 90° sú: **F**, **U**, **R**, **B**, **D** a **L**. Pri rotáciách v protismere hodinových ručičiek o 90° sa k písmenám pridáva apostrof. Rotácie v protismere teda vyzerajú takto: **F'**, **U'**, **R'**, **B'**, **D'** a **L'**. Keď je kocka otáčaná o 180° bez ohľadu na smer, lebo výsledná rotácia je rovnaká, pridáva sa k rotáciám číslo 2: **F2**, **U2**, **R2**, **B2**, **D2** a **L2**. Toto sú základné rotácie, ale existujú aj rotácie, ktoré sa používajú skôr v pokročilejších metódach: **M** (Middle layer) (otočenie strednej vrstvy kocky medzi stenami R a L, v rovnakom smere ako pohyb L), **E** (Equatorial layer) (otočenie rovníkovej vrstvy kocky medzi stenami D a U, v rovnakom smere ako pohyb D), **S** (Standing layer) (otočenie stojacej vrstvy kocky medzi stenami F a B, v rovnakom smere ako pohyb F), **x** (otočenie celej kocky v smere otáčania R), **y** (otočenie celej kocky v smere otáčania U), **z** (otočenie celej kocky v smere otáčania F) a **f**, **u**, **r**, **b**, **l**, **d** (taktiež značené ako **Fw**, **Uw**, **Rw**, **Bw**, **Lw**, **Dw**, kde spoločne so stenou sa otáča aj vrstva kocky pod ňou) (Obr. 21). [16][17]



Obr. 21: Rotácie kocky [17]

2.2 Začiatočnícka metóda (Beginner's method)

Riešenie začiatočníckej metódy spočíva v siedmych jednoduchých štádiách:

- 1) biely kríž,
- 2) biele rohy,
- 3) druhá vrstva,
- 4) žltý kríž,
- 5) žlté hrany,
- 6) žlté rohy,
- 7) orientácia žltých rohov.

Prvým štádiom je vytvorenie bieleho kríža na vrchnej strane kocky (Obr. 22). Dôležité je dbať na farby susedných stien a jednotlivé zdieľané hrany s hranami bielymi tak umiestňovať na správne pozície. Tento krok je však relatívne intuitívny a nie je potrebné dávať pozor na ostatné farby, okrem na farby patriace do kríža.



Obr. 22: Biely kríž

Druhým štádiom je správne orientovať a umiestniť biele rohy. Najprv je potrebné kocku otočiť hore nohami a aplikovať jednoduchý algoritmus, kde je premiestňovaný roh umiestnený tak, aby sa biely kríž nachádzal na spodnej strane kocky a premiestňovaný roh v pravom hornom rohu kocky. Následne sú vykonávané rotácie R , U , R' , U' , pokiaľ premiestňovaný roh nie je na svojom mieste a správne orientovaný. Tretím štádiom je dokončenie druhej vrstvy pomocou umiestnenia hrán na správne miesto vo vrstve. Pri tomto kroku je potrebné poznať dva algoritmy, tzv. Pravý a Ľavý algoritmus. Pri Pravom algoritme je premiestňovaná hrana umiestnená na hornú hranu prednej steny tak, aby farba hrany bola zhodná s farbou steny a susedná farba premiestňovanej hrany bola totožná s farbou

stény pravej. Keby sa farba premiestňovanej hrany zhodovala s farbou steny ľavej, išlo by o Ľavý algoritmus. Pravý algoritmus je potom $U, R, U', R' U' F', U, F$ a Ľavý algoritmus je $U', L', U, L, U, F, U', F'$. Keby sa premiestňovaná hrana už nachádzala na niektorej hrane vo vrstve druhej alebo bola nesprávne otočená, jedným z algoritmov by bolo možné hranu z daného miesta vybrať. Štvrtým štádiom je vytvorenie žltého kríža na vrchu kocky. V tomto kroku nie je potrebné brať ohľad na susedné farby hrán. Pri vytváraní kríža môžu nastať viaceré varianty, ktoré ale spolu súvisia. Prvý variant nastáva, keď sa na vrchnej stene nachádza iba žltá bodka v strede. Aplikovaný je algoritmus F, R, U, R', U', F' pričom vzniká vzor v tvare L na mieste kríža, rohy nie sú zatiaľ podstatné. Za potrebné je otočenie tvaru L do ľavého zadného rohu, podľa obrázka (Obr. 23). Keď sa nachádza L v rohu, je možné pokračovať už spomínaným algoritmom pre vytvorenie horizontálnej čiary a následne kríža.



Obr. 23: L v ľavom zadnom rohu

Piatym štádiom je umiestnenie hrán tak, aby sa farebne zhodovali s farbou steny. Keď sú prehodené farby hrán na susedných stenách, je potrebné otočiť kocku aby správne umiestnené hrany boli v stenách ľavá a zadná. Pokiaľ sú farby oproti sebe, je nutné tento krok vykonať dvakrát a to R, U, R', U, R, U^2, R' a U . Šiestym, predposledným krokom je umiestniť rohy na ich správne miesta, pričom nezáleží na orientácii. Keď sú rohy na správnom mieste, je možné tento krok preskočiť. V opačnom prípade je hľadaný aspoň jeden roh, ktorý je na správnej pozícii. Keď je taký roh nájdený, je nutné otočiť kocku tak, aby sa daný roh nachádzal v pravom vrchnom rohu kocky pri pohľade na kocku zo strany prednej. Aplikovaný je algoritmus $U, R, U', L', U, R', U' L$, pokiaľ nie je ani jeden z rohov na začiatku na správnej pozícii, je nutné tento krok vykonať dvakrát. Posledným, siedmym krokom je správne orientovať rohy vo vrchnej vrstve. Pri tomto kroku je nutné

byť opatrný, pretože sa kocka pri nesprávnom pohybe môže znovu zamiešať. Kocku otočíme tak, aby sa prvý nevyriešený roh nachádzal v pravom vrchnom rohu kocky pri pohľade na kocku zo strany prednej a aplikujeme algoritmus R', D', R, D , ktorý vykonávame, pokiaľ roh nie je umiestnený vo vrchnej vrstve správne. Pokračujeme otočením vrchnej steny aby sa na spomínanú pozíciu premiestnil ďalší nevyriešený roh, celá kocka nie je otáčaná. V priebehu tohto kroku sa môže zdať, že sa kocka znovu zamiešala, ale netreba mať obavy vopred, kocka sa znova zloží po správnej orientácii všetkých rohov. Pokiaľ boli všetky štádia vykonané správne, je kocka vyriešená. [18]

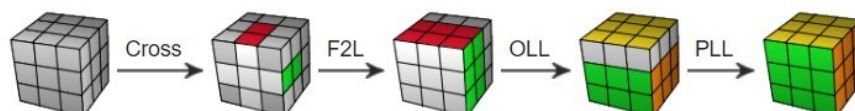
2.3 Pokročilá metóda (Advanced Method - CFOP)

Metóda CFOP alebo tiež označovaná ako Fridrichova metóda bola vytvorená viacerými riešiteľmi a pomenovaná v roku 1981 po česko-amerikej speedcuberke Jessica Fridrich (*Obr. 24*), ktorá patrila medzi riešiteľov. Pomohla tomu hlavne jej publikácia z roku 1995. [19]



Obr. 24: Jessica Fridrich [20]

Obdobný názov CFOP táto metóda získala zložením začiatkových písmen jednotlivých krokov - Cross, F2L (First two layers), OLL (Orient the last layer) a PLL (Permutate the last layer) (*Obr. 25*). [19]



Obr. 25: Jednotlivé kroky CFOP [21]

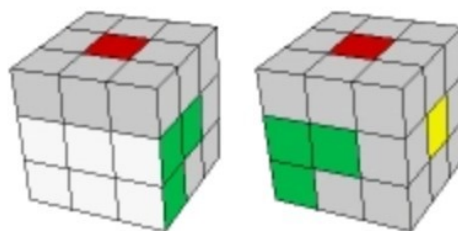
Podobne ako pri začiatkovej metóde, tak aj pri CFOP je prvým krokom vytvorenie bieleho kríža (Cross) na spodnej strane kocky. Riešenie kríža na spodnej strane je podstatné

pre zlepšenie celkového času potrebného na vyriešenie kocky, kedy už kocku nie je potreba otáčať, na rozdiel od začiatočnickej metódy, v snahe sa dostať pod 20 sekúnd. Stále to je však intuitívny krok, bez nutnosti poznať zložité algoritmy. Druhým krokom pri CFOP je riešenie prvých dvoch vrstiev súčasne (F2L), na rozdiel od začiatočnickej metódy, kde sa riešila prvá a druhá vrstva osobitne. Môže nastať 41 rôznych situácií, ktoré môžu byť riešené intuitívne, v prípade robota je lepšie používať tabuľku algoritmov. Princíp tohto kroku spočíva vo vytváraní párov vo vrchnej vrstve a následného vkladania páru na požadované miesto. Tretím krokom je riešenie orientácie komplementárnych farieb vo vrchnej, poslednej vrstve kocky (OLL). Nezáleží pritom na susedných farbách jednotlivých hrán a rohov, tie sú riešené v nasledujúcom kroku. Na vyriešenie tohto kroku je potrebné poznať 57 algoritmov, ktoré pokrývajú celú škálu možných situácií. Existuje aj jednoduchší variant, kde stačí 9 algoritmov, ale celkové riešenie je pomalšie. Jednoduchší variant sa označuje 2look OLL. Posledným, štvrtým krokom sú permutácie poslednej vrstvy (PLL). Riešenie tohto kroku spočíva v 21 rôznych algoritmoch, ale aj ako pri OLL existuje jednoduchší variant len so 6 algoritmami, ktorý je o pár sekúnd pomalší. Po zvládnutí týchto krokov je kocka vyriešená. [19][21]

2.4 Metóda Roux (Roux method)

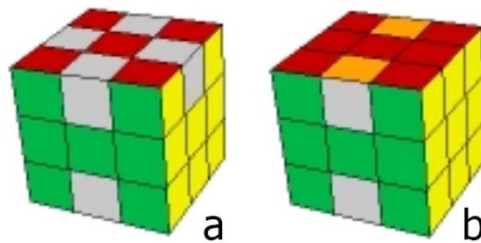
Metóda Roux bola vytvorená Francúzom Gilles Roux v roku 2003. Táto metóda sa stala veľkým konkurentom metódy CFOP, lebo hoci nie je veľmi používaná, ukázala svoj potenciál vo viacerých súťažiach. Zatiaľ čo metóda CFOP sa spolieha na prezieravosť s malým zameraním na intuitívne riešenie, v Roux dva hlavné kroky zahrňujú intuitívne vytváranie blokov. Taktiež Roux kladie dôležitosť na rýchlosti rotovaní kocky, kde napríklad v poslednom kroku používa len rotácie steny U a vrstvy M, ale aj na používaní pokročilých techník. [22]

Technika riešenia spočíva v prvom rade vo vytvorení bloku jeden štvorec široký, dva štvorce vysoký a tri štvorce hlboký na ľubovoľnom mieste na kocke (Obr. 26). [23]



Obr. 26: Blok 1x2x3 [23]

Jednoduché zvládnutie tohto kroku spočíva v dlhej praxi. Druhým krokom je vytvorenie druhého 1x2x3 bloku na opačnej strane vzhľadom k bloku prvému, bez jeho porušenia. Po vykonaní tohto kroku by mal vzniknúť jeden blok na pravej dolnej strane kocky a druhý na ľavej dolnej strane tak, aby stena U a vrstva M bola voľná. Tretím krokom je orientovanie a permutovanie 4 rohov v stene U (Obr. 27 a). Štvrtým krokom je orientovanie zvyšných 6 hrán, za použitia rotácií U a M. Krokom piatym je umiestnenie hrán do stien L a R, ktoré sú susediace so stenou U, výsledkom tohto kroku by mali byť správne vyriešené celé steny L a R (Obr. 27 b). Šiestym a posledným krokom je vyriešenie stredov a hrán vo vrstve M. [22][23]



Obr. 27: Roux – krok 3 a 5 [23]

2.5 Metóda ZZ (ZZ method)

Metódu ZZ vytvoril Zbigniew Zborowki v roku 2006. Táto metóda sa snaží skrátiť počet ťahov metódy CFOP, ale je zameraná aj na vysokú rýchlosť otáčania stien kocky. Na rozdiel od CFOP nepoužíva pohyby F, B alebo D pri prvom štádiu riešenia, čo znamená, že ruky riešiteľa neopúšťajú ľavú a pravú stranu kocky, a to vedie k rýchlejšiemu riešeniu. Poslednú vrstvu je potom možné vyriešiť na prvý pohľad, keďže všetky štyri hrany sú správne orientované, čo znamená menej prípadov riešenia. Avšak stále sa je potrebné naučiť 493 algoritmov, čo dokázalo len niekoľko desiatok ľudí v celom rozsahu. Jednoduchšie by bolo najprv použiť riešenie pomocou metódy ZZ a až pri poslednej vrstve kocku doriešiť pomocou metódy CFOP, čo by znamenalo 7 algoritmov pre jednoduché OLL namiesto 57. [24][25]

2.6 Metoda Thistlethwaite 45

Metoda Thistlethwaite alebo taktiež označovaná ako Thistlethwaite 45 bola vytvorená matematikom Morwenom B. Thistlethwaite v roku 1981. Ide o algoritmus, ktorý umožňuje vyriešiť Rubikovu kocku v pozoruhodne malom počte ťahov. Jedna literatúra uvá-

dza maximum 45 ťahov druhá zase 52. Avšak tento algoritmus je príliš komplikovaný na to, aby si ho ľudia mohli zapamätať, a preto je prakticky len pre počítače. Dôležitý je však z teoretického hľadiska, pretože dlhú dobu táto metóda bola metódou s najmenším počtom ťahov. [26][27]

Algoritmus sa od ostatných líši tým, že neumiestňuje kúsky na ich správne pozície jeden po druhom, ale vytvára menšie a menšie skupiny, pri ktorých pracuje so všetkými zároveň pokiaľ neostane pre každý kúsok len jedna možná pozícia a kocka nebude vyriešená. Spôsob riešenia spočíva v počiatočnej sérii pohybov až kým nevznikne taká kombinácia, ktorú je možné riešiť bez použitia štvrtinových otočiek (otočenie o 90°) stien U a D. Pokračuje v riešení kocky tak, že nájde kombináciu, kde ani steny F a B nepotrebujú štvrtinové otočky. S týmito ďalšími obmedzeniami sa dospeje ku kombinácii, kde pre vyriešenie kocky nie sú potrebné žiadne štvrtinové otočky a kocka môže byť vyriešená iba polovičnými otočkami (otáčanie o 180°). Tieto štyri štádia sú dosť komplikované, lebo využívajú veľké vyhľadávacie tabuľky pre všetky pozície v každej fáze. [28]

2.7 God's Number

God's Number v terminológii klasickej Rubikovej kocky predstavuje maximálny potrebný počet pohybov na vyriešenie každej z 43 252 003 274 489 856 000 možných kombinácií. Bolo dokázané, že týmto číslom je číslo 20. Na prvý pohľad sa môže zdať, že 20 potrebných pohybov je málo, ale teoreticky potrebných je ešte menej. Iba okolo 490 000 000 možných kombinácií vyžaduje 20 pohybov na vyriešenie kocky. Hoci 490 miliónov je obrovské číslo, je stále zlomkom z 43 triliónov možných kombinácií. Presne percentuálne vyjadrené na 0.0000011328955%. [29][30]

Nasledujúci obrázok (*Obr. 28*) ukazuje pomer medzi počtom pohybov potrebných na vyriešenie Rubikovej kocky k počtu rôznych pozícií, ktoré môžu nastať.

Distance	Count of Positions
0	1
1	18
2	243
3	3,240
4	43,239
5	574,908
6	7,618,438
7	100,803,036
8	1,332,343,288
9	17,596,479,795
10	232,248,063,316
11	3,063,288,809,012
12	40,374,425,656,248
13	531,653,418,284,628
14	6,989,320,578,825,358
15	91,365,146,187,124,313
16	about 1,100,000,000,000,000,000
17	about 12,000,000,000,000,000,000
18	about 29,000,000,000,000,000,000
19	about 1,500,000,000,000,000,000
20	about 490,000,000

Obr. 28: Pomer medzi počtom pohybov a počtom pozícií [29]

2.7.1 The Super-Flip

Najznámejšie zamiešanie kocky, ktoré potrebuje na vyriešenie presne 20 pohybov sa anglicky nazýva The Super-Flip (*Obr. 29*). Na vytvorenie tohto konkrétneho zamiešania je potrebné kocku rotovať z akejkoľvek počiatočnej orientácie pomocou pohybov: R, L, U₂, F, U⁺, D, F₂, R₂, B₂, L, U₂, F⁺, B⁺, U, R₂, D, F₂, U, R₂, U, kde pohyby U₂, F₂, R₂, B₂ a R₂ sú považované za jeden pohyb. Toto zamiešanie, ako je na obrázku viditeľné, predstavuje zamiešanie kde každý roh je na svojom mieste správne orientovaný a každá hrana je na svojom mieste ale orientovaná opačne. [30]



Obr. 29: The Super-Flip

3 ROBOTIKA V PRIEMYSLE

Použitie robotických systémov v priemysle nie je najnovší pojem. Avšak čoraz viac sa je možné stretnúť s robotom na pozícii, na ktorej by ho len málo kto očakával.

3.1 ABB

Spoločnosť ABB je svetovo poprednou vo výkonových a automatizačných technológiách. Zamestnáva približne 110 000 ľudí a je aktívna približne v 100 krajinách po celom svete. Rozmiestnenie prevádzok zabezpečuje bezproblémový a rýchly kontakt so zákazníkmi nech sú kdekoľvek. [31]

3.1.1 O spoločnosti ABB

Spoločnosť ABB ako ju dnes poznáme vznikla v roku 1988, avšak jej historické začiatky sa evidujú až do 19. storočia, kedy vznikla spojením spoločností ASEA a BBC. Novovzniknutá spoločnosť sa usadila vo Švajčiarsku v meste Zürich a zamestnávala vyše 160 000 ľudí po celom svete. V roku 1990 spoločnosť ABB uviedla na trh Azipod, skupinu elektrických pohonných systémov, ktoré sú pripevnené na vonkajšej strane lodí. Zvyšujú výkonnosť, manévrovateľnosť a priestor na palube. V roku 1998 ABB uviedla FlexPicker, delta robot špeciálne navrhnutý pre aplikácie odoberania a balenia v priemysle. Dnes je lídrom vo svojej oblasti. V roku 2000 dodala prvé komerčné využitie vedenia vysokého napätia medzi loďou a pevninou. Vďaka tejto dodávke sa znížili emisie skleníkových plynov z lodí kotviacich vo švédskom prístave Göteborg. V roku 2004 uviedla rozšírený automatizovaný systém 800xA, poskytujúci systém automatizácie procesov, ktorý bol od svojho vzniku nainštalovaný v tisíckach spracovateľských fabriek. V roku 2010 spojila vodnú elektrárňu Xiangjiaba s mestom Šanghaj vzdialenom približne 2 000 km. V roku 2014 spoločnosť ABB predstavila prvého skutočne kolaboratívneho robota na svete YuMi. Inovatívny dvojramenný robot s prelomovou funkcionalitou odhaľuje obrovský potenciál v globálnej automatizácii v priemysle. V roku 2018 sa spoločnosť pripojila k majstrovstvám FIA Formule E ako titulný partner pre vytvorenie novopomenovaných majstrovstiev ABB FIA Formule E. Preteky slúžili ako konkurenčná platforma pre testovanie a vývoj technológií súvisiacich s elektronickou mobilitou. [32]

3.1.2 IRB 14000 YuMi

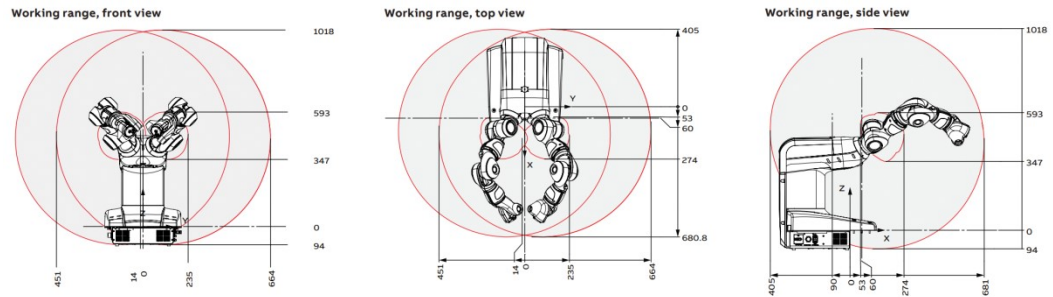
Skutočný kolaboratívny dvojramenný robot YuMi, alebo celým menom IRB 14000 YuMi (Obr. 30), už ako bolo spomínané, bol prvýkrát predstavený 9. septembra 2014 spoločnosťou ABB. YuMi bol v prvom rade vyvinutý tak, aby vyhovoval flexibilným a obratným potrebám vo výrobe spotrebného elektronického priemyslu. YuMi bol navrhnutý so schopnosťou vidieť a cítiť. Mäkké čalúnenie konštrukcie ramien v kombinácii s inovatívnymi technológiami snímania sily zabezpečuje bezpečnosť ľudských spolupracovníkov. Bezpečnosť je zabudovaná do funkčnosti samotného robota, takže môže pracovať mimo kletky. Presnosť robota je jedným zo známych faktorov, keďže je prispôbený pre zvládnutie manipulácie s krehkými súčiastkami v hodinkách, ale aj komponentmi používanými v mobilných telefónoch, tabletoch a stolových počítačoch. [33]



Obr. 30: IRB 14000 YuMi [34]

3.1.2.1 Technické parametre

Robot IRB 14000 YuMi zvládne uniesť záťaž 0,5 kg a jeho dosah je 0,559 m. Robot disponuje 14 osami (7 v každom ramene) a krytím IP30. Montážne je usposobený pre montáž na stôl alebo iný rovný povrch. Taktiež disponuje integrovaným ovládačom, integrovaným nasávaním vzduchu predpripraveným pre koncový efektor so sacou silou 4 Bar a integrovaným ethernetom 100/10 Base-TX. Maximálna rýchlosť ramena je 1,5 m/s so zrýchlením 11 m/s². Fyzické rozmery základne robota sú 399x496 mm. Hmotnosť robota je 38 kg. Pracovný priestor je vyobrazený na obrázku (Obr. 31). [35]



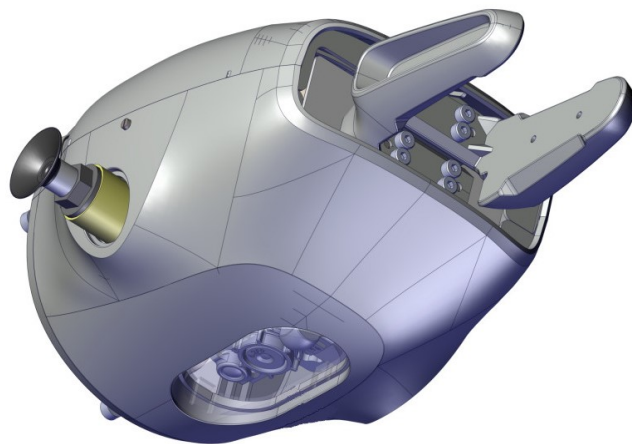
Obr. 31: Pracovný priestor robota IRB 14000 YuMi [35]

3.1.2.2 SmartGripper

Uchopovač SmartGripper (Obr. 32) sa vyrába vo viacerých variantoch:

- servo modul
- servo modul + jeden sací modul
- servo modul + dva sacie moduly
- servo modul + jeden modul videnia
- servo modul + jeden modul videnia + jeden sací modul.

Čeľusť uchopovača je tvorená dvoma prstami a je ju možné otvoriť do dĺžky 0-50 mm (max. 25 mm pre prst) s rýchlosťou 25 mm/s. Maximálna sila uchopenia je 20 N. Sací modul je navrhnutý na maximálne užitočné zaťaženia 150 g, ale táto skutočnosť závisí od viacerých faktorov: dizajn prísavky, štruktúra povrchu objektu, pohyb robota s prisatým objektom. Vizuálny modul obsahuje kameru AE3 od spoločnosti Cognex s rozlíšením 1,3 Megapixelu, ktorá poskytuje výkonné a spoľahlivé videnie určené pre identifikáciu. [36]



Obr. 32: SmartGripper [36]

3.1.2.3 Spôsob komunikácie

IRB 14000 YuMi ako spôsob komunikácie k reálnemu systému môže použiť socketovú TCP/IP komunikáciu alebo aj OPC protokol. Klasické priemyselné zbernice (ProfiNet, Ethernet/IP) pre komunikáciu v tomto prípade nie sú vhodné, z hľadiska absencie podpory vo virtuálnom prostredí.

Prostredníctvom socketovej komunikácie TCP/IP si je robot IRB 14000 YuMi schopný vymieňať informácie o procese (spúšť kamery, farby na jednotlivých pozíciách), čo nie je zložité na implementáciu. YuMi disponuje čiernobiou kamerou umiestnenou v uchopovači, ktorá je napriamo spojená cez Integrated Vision. Pomocou relé, cez digitálne výstupy robota, je možné ovládanie RGB prísvitú kocky pre získanie jednotlivých farieb z jasú čiernobielych fotiek. Integrated Vision obsahuje kompletné softvérové a hardvérové riešenie, ktoré je plne integrované s riadiacou jednotkou IRC5 a programovacím prostredím RobotStudio. Viditeľnosť je založená na rade inteligentných kamier Cognex, so zabudovaným spracovaním obrazu a komunikačným rozhraním Ethernet. [37]

OPC protokol je rad noriem a špecifikácií pre priemyselnú telekomunikáciu, čoho súčasťou je komunikačný protokol OPC-UA (Unified Architecture), ktorý vznikol ako odozva na zjednotenie používaných protokolov naprieč výrobcami komponentov pre automatizáciu a robotizáciu. Komunikácia OPC-UA prostredníctvom TSN, „OPC-UA over TSN“ (aktuálne vo vývoji), by mala pridávať možnosť komunikácie v reálnom čase medzi zariadeniami, ktoré vyžadujú rýchlu odozvu. [38]

3.1.3 RobotStudio

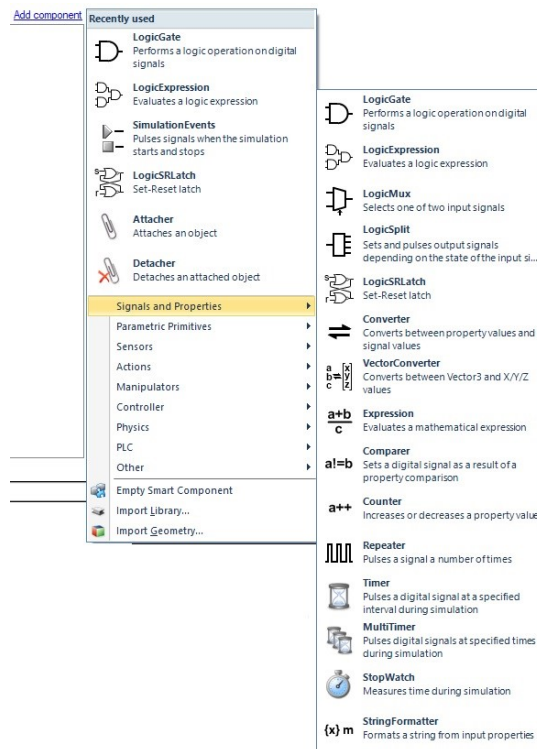
RobotStudio od ABB je simulačné a programovacie softvérové prostredie. Ponúka kompletnú virtuálizáciu reálnych systémov, čo zabezpečuje napríklad možnosť sledovania výrobnéj linky na diaľku bez potreby fyzickej návštevy alebo preprogramovanie robotov bez nutnosti zastavenia výrobnéj linky. Takýto prístup sa dá eventuálne označiť ako digitálne dvojča. Použitie programovacieho nástroja RobotStudio v režime offline umožňuje užívateľom vytvárať, simulovať a testovať kompletnú inštaláciu robotov vo virtuálnom 3D prostredí. [39]

3.1.3.1 Smart Componenty

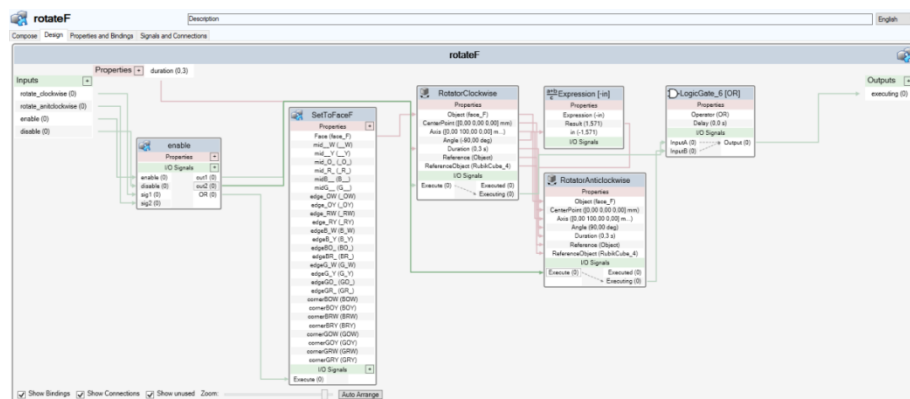
Smart Componenty sú objekty RobotStudia so zabudovanými vlastnosťami a logikou pre simulovanie komponentov, ktoré nie sú súčasťou virtuálneho radiča. RobotStudio ponúka

sadu základných Smart Componentov pre základné pohyby, signálnu logiku, aritmetiku, parametrické modelovanie, senzory, atď. Základné Smart Componenty sa dajú kombinovať pre vytvorenie komponentov s komplexnejšími vlast'ami. Napríklad môže ísť o pohyby Gripera (uchopovača), ale aj o predmety pohybujúce sa po dopravníkoch. Komplexné komponenty sú tvorené cez grafické programovacie rozhranie a môžu byť súčasťou staníc a simulácii všetkých možných scenárov.

Vytváranie, upravovanie a agregovanie Smart Componentov prebieha pomocou grafického používateľského rozhrania. V kontextovom okne (Obr. 33) je možné z kategórií vybrať jednotlivé Smart Componenty. Obrázok (Obr. 34) prezentuje použité v praxi.



Obr. 33: Kontextové okno komponentov



Obr. 34: Použitie Smart Componentov v praxi

3.1.4 RAPID

RAPID je programovací jazyk vysokej úrovne používaný na riadenie priemyselných robotov od spoločnosti ABB. Bol zavedený v roku 1994, keď nahradil predchádzajúci programovací jazyk ARLA.

Medzi hlavné funkcie jazyka patria:

- Rutinné parametre
 - procedúry – sú používané ako podprogramy
 - funkcie – vracajú hodnotu konkrétneho typu a používajú sa ako argument inštrukcie
 - trap rutiny – sú prostriedkom reakcie na prerušenie
- aritmetické a logické výrazy
- automatické spracovanie chýb
- modulárne programy
- multitasking [40]

3.2 Digitálne dvojča

Digitálne dvojča predstavuje virtuálnu (softvérovú) repliku fyzického produktu (zariadenia) alebo procesu, ktorá sa používa na porozumenie a predpovedanie výkonnostných charakteristík fyzického zariadenia. Digitálne dvojča sa používa na spustenie simulácií pred zostavením a nasadením skutočných zariadení, ale aj počas celého životného cyklu pre simuláciu, predpovedanie a optimalizáciu technológií. [41][42]

II. PRAKTICKÁ ČASŤ

4 IMPLEMENTOVANIE ALGORITMU DO KÓDU

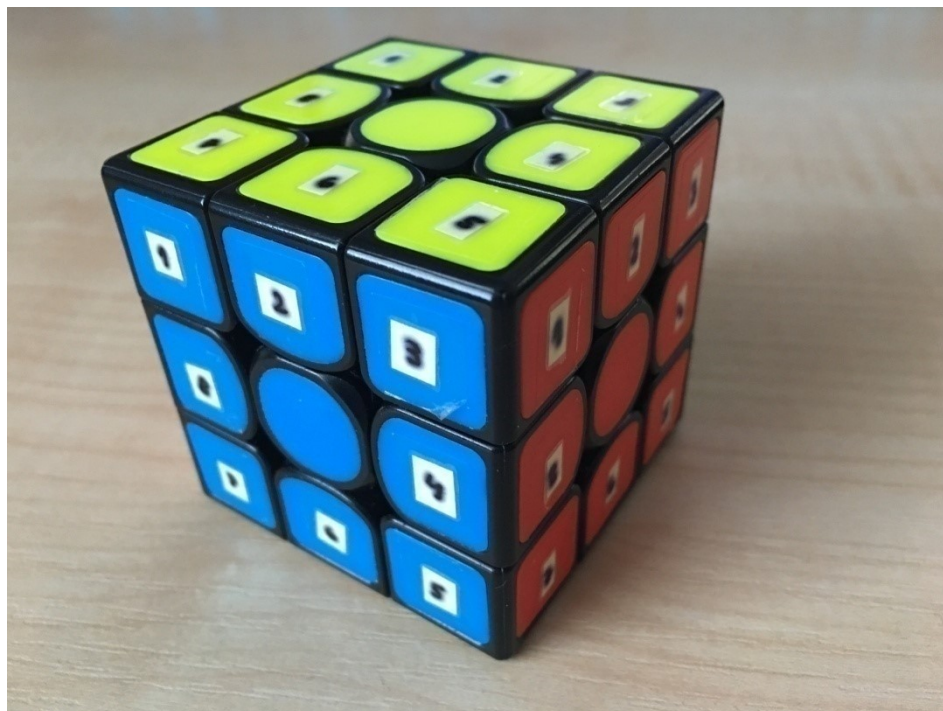
Prvou dôležitou súčasťou praktickej časti tejto práce bolo implementovanie zvoleného algoritmu do kódu.

4.1 Zápis kocky

Poskladaná kocka je v kóde reprezentovaná ako dvojrozmerné pole s rozmerom 6x9 prvkov, 6 farebných stien kocky a 9 štvorcových kúskov na každej stene. Jednotlivé farebné steny a štvorcové kúsky (ďalej len „štvorce“) sú prezentované číselne kvôli abstraktnejšiemu vyjadreniu jednotlivých farieb.

```
CONST byte solved{6,9} := [[1,1,1,1,1,1,1,1,1],[2,2,2,2,2,2,2,2,2],[3,3,3,3,3,3,3,3,3],  
                          [4,4,4,4,4,4,4,4,4],[5,5,5,5,5,5,5,5,5],[6,6,6,6,6,6,6,6,6]];
```

Finálne navrhnuté riešenie kocky nie je závislé na počiatočnom otočení poskladanej kocky, to znamená, že kocka môže byť otočená napríklad žltou farbou nahor, ale to vôbec nie je podmienkou. Hore môže byť aj iná z farieb. Toto je umožnené vďaka závislosti algoritmu k stenám podľa umiestnenia v priestore, kde predná stena je značená ako F (front) a v kóde ju prezentuje číslo 1. Obdobne sú značené ostatné steny: U je 2, R je 3, B je 4, D je 5 a nakoniec L je 6. Takéto značenie bolo navrhnuté len na osobnej preferencii a nedá sa povedať, že existuje nejaké vopred dohodnuté číselné označenie vo svete. Taktiež bolo navrhnuté označenie jednotlivých štvorcov v spomínaných stenách. Ako praktické sa osvedčilo značenie do tzv. hada, ktoré začína v ľavom hornom rohu a končí v strede z dôvodu umiestnenia posledného štvorca, ktorý reprezentuje aj danú farebnú stenu. Taktiež výhodou použitého označenia je reprezentácia hrán párnymi číslami a rohy so stredom nepárnymi. Keďže sa poloha stredových štvorcov pri rotácii nemení, bude vždy na deviatom (poslednom) mieste vnútorného poľa. Pre lepšiu predstavu slúžia nasledujúce obrázky (*Obr. 35*), (*Obr. 36*).



Obr. 35: Reálna kocka olepená číslami pre lepšiu vizualizáciu

			(2,1)	(2,2)	(2,3)															
			(2,8)	(2,9)	(2,4)															
			(2,7)	(2,6)	(2,5)															
(6,1)	(6,2)	(6,3)	(1,1)	(1,2)	(1,3)	(3,1)	(3,2)	(3,3)	(4,1)	(4,2)	(4,3)									
(6,8)	(6,9)	(6,4)	(1,8)	(1,9)	(1,4)	(3,8)	(3,9)	(3,4)	(4,8)	(4,9)	(4,4)									
(6,7)	(6,6)	(6,5)	(1,7)	(1,6)	(1,5)	(3,7)	(3,6)	(3,5)	(4,7)	(4,6)	(4,5)									
			(4,1)	(4,2)	(4,3)															
			(4,8)	(4,9)	(4,4)															
			(4,7)	(4,6)	(4,5)															

Obr. 36: Rozložená kocka s použitým označením

4.2 Rotácie kocky

Jednotlivé rotácie kocky sú zapísané cez procedúry. Keďže je 6 stien a každú stenu je možné otáčať v smere alebo proti smeru hodinových ručičiek, dostávame celkovo 12 procedúr pre rotácie. V kóde sú tieto procedúry značené podľa jednoduchých pravidiel. Prvá časť názvu procedúry bude začínajúce písmeno jednej zo stien, napríklad predná, čiže F (front). Za toto reprezentujúce písmeno bol pridaný smer otáčania, značený z anglického jazyka, a to buď v smere hodinových ručičiek CW (clockwise) alebo protismeru hodinových ručičiek CCW (counter clockwise). Dodržaním týchto pravidiel bol získaný názov procedúry, napríklad *F_CW* alebo *F_CCW*, ktorá poskytuje informáciu o stene, ktorá je rotovaná a taktiež o smere, v ktorom je rotovaná. Všetky rotácie sú o 90°.

Pre lepšie vysvetlenie bola použitá predná stena rotovaná v smere hodinových ručičiek (*F_CW*).

```
PROC F_CW()
  VAR byte temp;

  temp := cube{1,1};
  cube{1,1} := cube{1,7};
  cube{1,7} := cube{1,5};
  cube{1,5} := cube{1,3};
  cube{1,3} := temp;

  temp := cube{1,2};
  cube{1,2} := cube{1,8};
  cube{1,8} := cube{1,6};
  cube{1,6} := cube{1,4};
  cube{1,4} := temp;

  temp := cube{2,7};
  cube{2,7} := cube{6,5};
  cube{6,5} := cube{5,3};
  cube{5,3} := cube{3,1};
  cube{3,1} := temp;

  temp := cube{2,6};
  cube{2,6} := cube{6,4};
  cube{6,4} := cube{5,2};
  cube{5,2} := cube{3,8};
  cube{3,8} := temp;

  temp := cube{2,5};
  cube{2,5} := cube{6,3};
  cube{6,3} := cube{5,1};
  cube{5,1} := cube{3,7};
  cube{3,7} := temp;

  update_solution "F";
ENDPROC
```

Vhodné je podotknúť, že pri otáčaní jednej zo stien sa otáčajú aj krajné (susedné) štvorce vedľajších stien, ktorým v kóde zodpovedajú odseky 3 až 5.

Pri riešení rotácie v protismere hodinových ručičiek, ako už plyní z názvu, sa mení smer premiestňovania farieb do protismeru. Napríklad pri rotovaní prednej steny o 90° v protismere hodinových ručičiek (*F_CCW*) sú premiestňované farby na pozíciu (1,1) (ľavý horný roh) z pozície (1,3) (pravý horný roh). Rovnako ako pri premiestňovaní v smere hodinových ručičiek, aj tu sú premiestňované krajné (susedné) štvorce vedľajších stien.

Podobným postupom sú riešené aj ostatné rotácie, či už v smere hodinových ručičiek alebo v protismere.

Poslednou súčasťou rotačných procedúr je procedúra *update_solution*, ktorá ukladá vykonané pohyby do poľa s riešením.

4.3 Riešenie kríža na spodnej strane kocky

Riešenie kocky pri začiatocníkovej a pokročilej metóde, ako vyplýva aj z teoretickej časti, sa väčšinou začína poskladaním kríža na spodnej strane kocky. Veľmi podstatné je skladať kríž tak, aby už poskladané časti vo finále nezmenili svoju polohu a dôležité je dbať aj na správne umiestnenie vzhľadom na krajné (susedné) steny, keďže ide o hranu, ktorá má 2 farby.

Celé riešenie kríža v kóde reprezentuje procedúra s názvom *cross_down*, ktorej súčasťou je funkcia *WHILE*, obsahujúca podmienku pre vykonávanie cyklu pokiaľ nebudú všetky kocky na správnej pozícii. Takéto riešenie je nutné kvôli premiestňovaniu farebných kociek v ostatných vrstvách, čo môže zapríčiniť umiestnenie hrany na miesto, ktorým program už prešiel.

```
WHILE cube{5,2} <> cube{5,9} OR cube{1,6} <> cube{1,9} OR
      cube{5,4} <> cube{5,9} OR cube{3,6} <> cube{3,9} OR
      cube{5,6} <> cube{5,9} OR cube{4,6} <> cube{4,9} OR
      cube{5,8} <> cube{5,9} OR cube{6,6} <> cube{6,9} DO
```

Podrobnejšie je táto podmienka testovaná formou vykonávaj cyklus až do chvíle, kým kocka zo steny 5 (D) nemá na pozícii 2 (hrana hore) v príslušnej stene farbu, ktorá sa nachádza na stene 5 (D) a pozícii 9 (stred) a zároveň kocka zo steny 1 (F) nemá na pozícii 6 (hrana dole) v príslušnej stene farbu, ktorá sa nachádza na stene 1 (F) a pozícii 9 (stred). Takto riešená podmienka zaisťuje správnosť umiestnenia a orientácie hrán. Rovnakým princípom sú riešené ostatné steny ako plyní z úseku kódu nad textom.

4.3.1 Prípád umiestnenia hľadanej hrany dole, orientovanej správne na správnej alebo nesprávnej pozícií

Najjednoduchší prípad môže nastať, keď sa hľadaná hrana nachádza dole, správne orientovaná na správnej alebo nesprávnej pozícií.

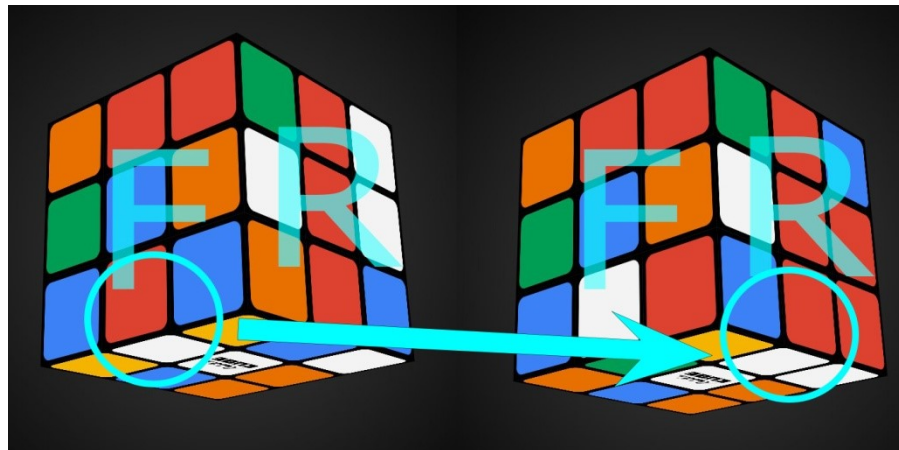
```

IF cube{5,2} = cube{5,9} THEN
  IF cube{1,6} = cube{1,9} THEN
    !do nothing
  ELSEIF cube{1,6} = cube{3,9} THEN
    D_CW;
    R_CW;
    D_CCW;
    R_CCW;
  ELSEIF cube{1,6} = cube{4,9} THEN
    D_CW;
    D_CW;
    B_CW;
    D_CCW;
    D_CCW;
    B_CCW;
  ELSEIF cube{1,6} = cube{6,9} THEN
    D_CCW;
    L_CW;
    D_CW;
    L_CCW;
  ENDF
ENDIF

```

Prvý *IF* testuje, či sa na stene 5 (D) na pozícií 2 (hrana hore) nachádza farba zhodná s farbou na tej istej stene 5 (D) na pozícií 9 (stred). Pokiaľ je táto podmienka splnená, testujú sa vnútorné podmienky. Keď je na stene 1 (F) a pozícií 6 (hrana dole) farba zhodná s farbou na tej istej stene 1 (F) pozícií 9 (stred), nevykonávajú žiadne pohyby, kúsok je na správnom mieste. V opačnom prípade sa testuje ďalší *IF* (*ELSEIF*). Keď je na stene 1 (F) a pozícií 6 (hrana dole) farba zhodná s farbou na stene 3 (R) pozícií 9 (stred) (*Obr. 37*), hrana sa nenachádza na správnej pozícií, je za potrebné vykonať nasledujúce rotácie:

- otočenie spodnej steny kocky v smere hodinových ručičiek (*D_CW*),
- otočenie pravej steny kocky v smere hodinových ručičiek (*R_CW*),
- otočenie spodnej steny kocky v protismere hodinových ručičiek (*D_CCW*),
- otočenie pravej steny kocky v protismere hodinových ručičiek (*R_CCW*).



Obr. 37: Presun hrany zo stien FD do RD, orientovanej správne

Teraz sa hrana nachádza na správnom mieste a žiadna iná, už správne umiestnená hrana v spodnej vrstve, nezmenila svoje umiestnenie. Takýmto spôsobom sa testujú aj ostatné podmienky, či už na pozícií (1,6) alebo (3,6), (4,6) a (6,6) (vnútorné podmienky *IF*), ale aj na pozíciách (5,2), (5,4), (5,6) a (5,8) (vonkajšie podmienky *IF*).

4.3.2 Prípád umiestnenia hľadanej hrany hore, orientovanej správne

Pri umiestnení hľadanej hrany hore, keďže je orientovaná správne, ide principiálne o to isté ako pri umiestnení dole, s jedinou výnimkou hľadania požadovanej hrany, patriacej do steny 5 (D), na stene 2 (U), ako už plynie aj z názvu.

```

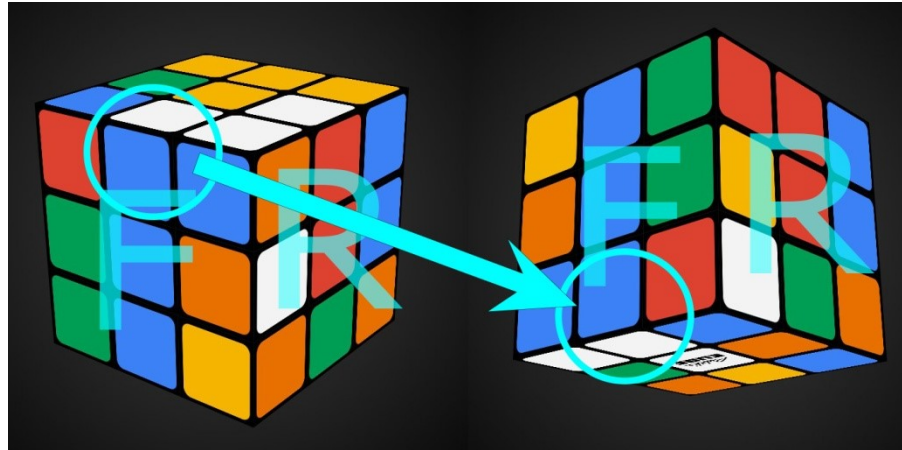
IF cube{2,6} = cube{5,9} THEN
  IF cube{1,2} = cube{1,9} THEN
    F_CW;
    F_CW;
  ELSEIF cube{1,2} = cube{3,9} THEN
    U_CCW;
    R_CW;
    R_CW;
  ELSEIF cube{1,2} = cube{4,9} THEN
    U_CW;
    U_CW;
    B_CW;
    B_CW;
  ELSEIF cube{1,2} = cube{6,9} THEN
    U_CW;
    L_CW;
    L_CW;
  ENDF
ENDIF

```

Testuje sa, či sa na stene 2 (U) na pozícií 6 (hrana dole) nachádza farba zhodná s farbou na stene 5 (D) na pozícií 9 (stred). Pokiaľ je podmienka splnená, testujú sa vnútorné podmienky. Keď je na stene 1 (F) a pozícií 2 (hrana hore) farba zhodná s farbou na tej istej

stene 1 (F) a pozícii 9 (stred) (*Obr. 38*), hľadaná hrana sa nenachádza na správnej pozícii, je potrebné vykonať nasledujúce pohyby:

- otočenie prednej steny kocky v smere hodinových ručičiek (F_CW),
- otočenie prednej steny kocky v smere hodinových ručičiek (F_CW).



Obr. 38: Presun hrany zo stien FU do FD, orientovanej správne

Sú to vlastne dva rovnaké pohyby, ktoré vo finále zarotujú prednú stenu o 180° . Takýmto pohybom docielime umiestnenie hľadanej hrany na správnu pozíciu. Obdobne sa testujú a riešia aj ostatné pozície.

4.3.3 Prípady umiestnenia hľadanej hrany v jednej zo stien

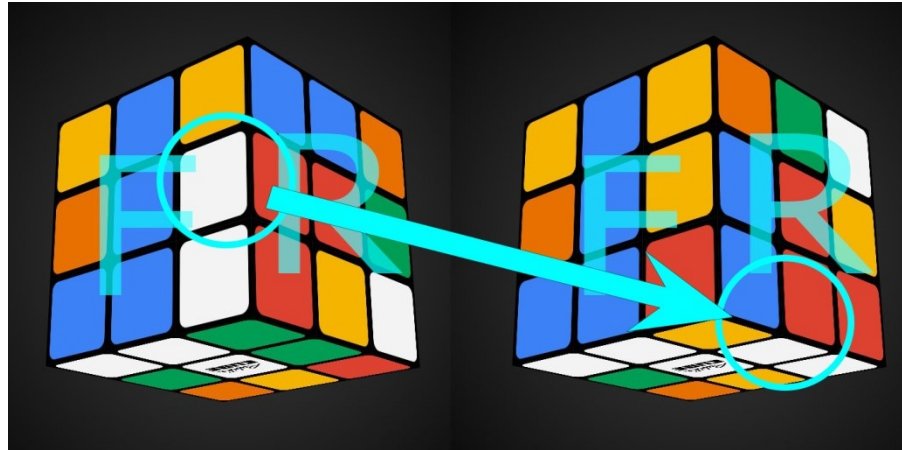
Pri umiestnení hľadanej hrany v jednej zo stien môže nastať viacero prípadov. Najjednoduchší prípad je, keď sa hľadaná hrana nachádza v príslušnej stene, ktorú stačí rotovať v jednom zo smerov.

```

IF cube{1,4} = cube{5,9} THEN
  IF cube{3,8} = cube{3,9} THEN
    R_CCW;
  ENDIF
ENDIF

```

Napríklad hrana, ktorej prvý štvorec je na pozícii (3,8) a má farbu zhodnú so stredom, ktorý reprezentuje danú stenu (3,9), jeho druhý štvorec sa nachádza na pozícii (1,4), ktorého farba je zhodná s farbou reprezentujúcou spodnú stenu (5,9) (*Obr. 39*), stačí rotovať o 90° v protismere hodinových ručičiek, aby sa dostal na požadovanú pozíciu.



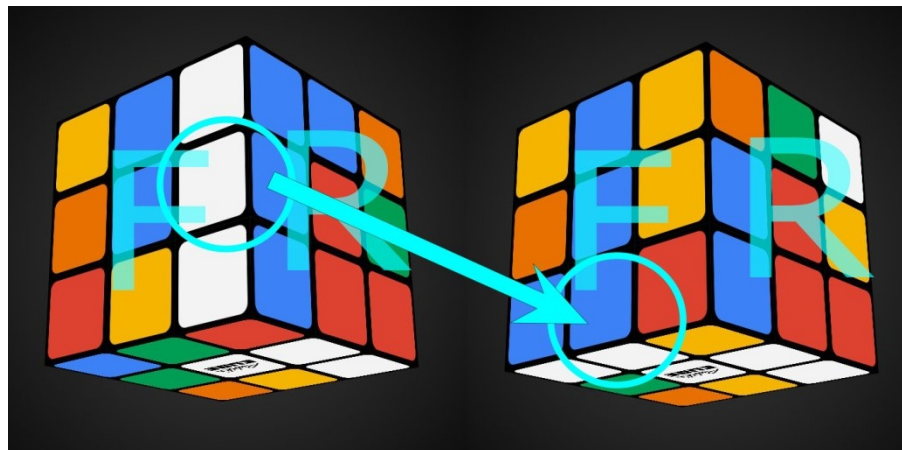
Obr. 39: Presun hrany zo stien FR do RD

Keď sa hľadaná hrana nachádza v jednej z príslušných stien a je potrebné ju premiestniť o jednu stenu ďalej (Obr. 40), použitá je jednoduchá kombinácia pohybov, pričom je rotovaná spodná stena spoločne so stenou, v ktorej sa premiestňovaná hrana nachádza.

```

IF cube{1,4} = cube{5,9} THEN
  IF cube{3,8} = cube{1,9} THEN
    D_CW;
    R_CCW;
    D_CCW;
  ELSEIF cube{3,8} = cube{4,9} THEN
    D_CCW;
    R_CCW;
    D_CW;
  ENDIF
ENDIF
ENDIF

```



Obr. 40: Presun hrany zo stien FR do FD

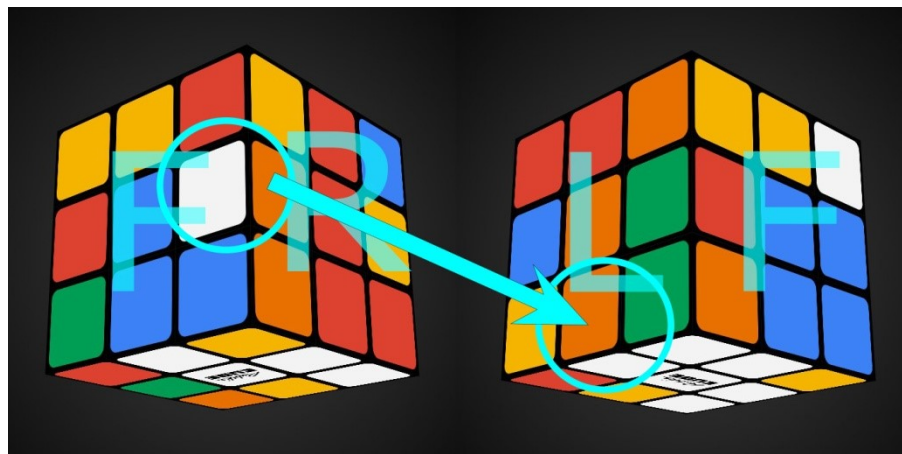
Pri premiestňovaní hľadanej hrany na ľavú stenu je otáčaná najprv spodná stena v smere hodinových ručičiek, potom stena, v ktorej sa hľadaná hrana nachádza a nakoniec znova spodná stena, ale v protismere hodinových ručičiek. Pri premiestňovaní na pravú stenu je potrebné vymeniť poradie otáčania spodnej steny.

Posledný pohyb z prípadu, keď sa hľadaná hrana nachádza na jednej z príslušných stien a je ho potrebné premiestniť na protiľahlú stenu (*Obr. 41*), sa vykonáva podobne ako pri premiestnení o jednu stenu, s pridaním dvojitej rotácie v spodnej stene (nezáleží na smere).

```

IF cube{1,4} = cube{5,9} THEN
  IF cube{3,8} = cube{6,9} THEN
    D_CW;
    D_CW;
    R_CCW;
    D_CW;
    D_CW;
  ENDIF
ENDIF
ENDIF

```



Obr. 41: Presun hrany zo stien FR do LD

4.3.4 Prípád umiestnenia hľadanej hrany dole

Ďalší prípad nastáva, keď sa prvý štvorec hrany nachádza na pozícii 6 (hrana dole), na jednej z možných susedných stien so stenou 5 (D), a jeho druhý štvorec sa nachádza na stene 5 (D) na jednej z možných pozícií.

```

IF cube{1,6} = cube{5,9} THEN
  IF cube{5,2} = cube{3,9} THEN
    F_CCW;
    R_CCW;
  ELSEIF cube{5,2} = cube{6,9} THEN
    F_CW;
    L_CW;
  ENDIF
ENDIF
ENDIF

```

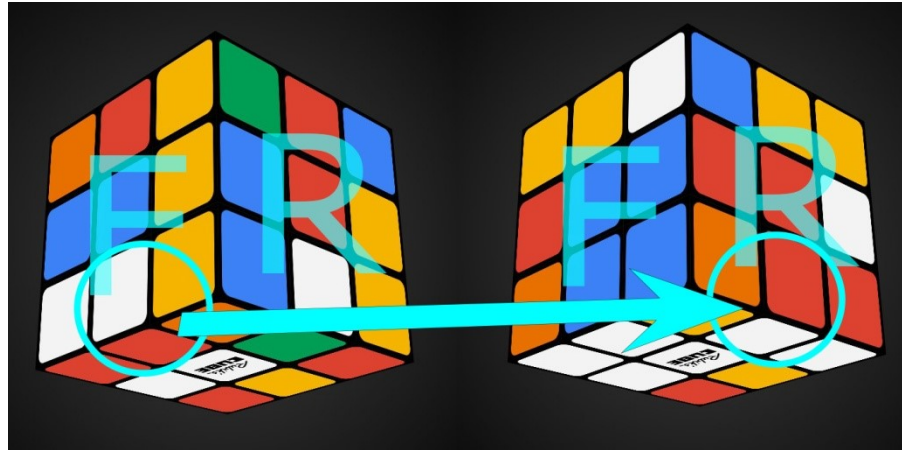
Napríklad prvý štvorec hrany je na pozícii (1,6) hľadanej farby (5,9) a jej druhá časť na pozícii (5,2) s farbou (3,9) (*Obr. 42*) alebo (6,9) (*Obr. 43*). Aby sa tieto hľadané hrany dostali na správnu pozíciu, je potrebné o 90° rotovať:

- prednú stenu kocky v protismere hodinových ručičiek,

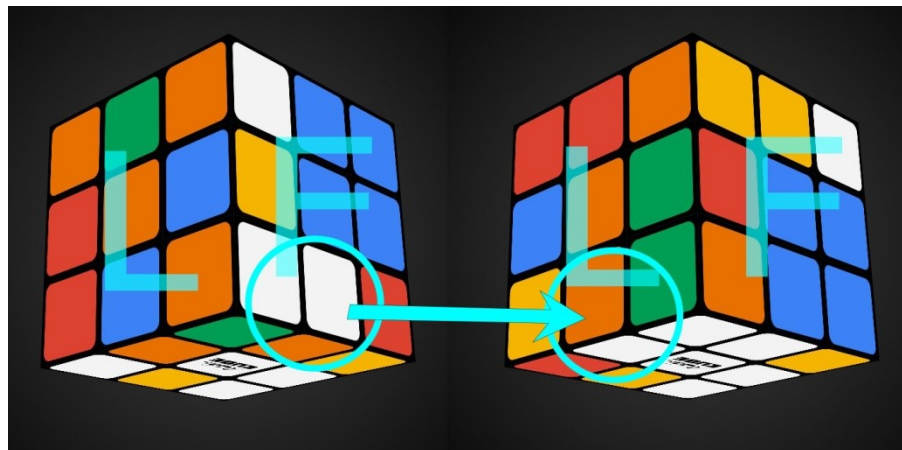
- pravú stenu kocky v protismere hodinových ručičiek (*Obr. 42*)

alebo rotovať

- prednú stenu kocky v smere hodinových ručičiek,
- ľavú stenu kocky v smere hodinových ručičiek (*Obr. 43*).



Obr. 42: Presun hrany zo stien FD do RD



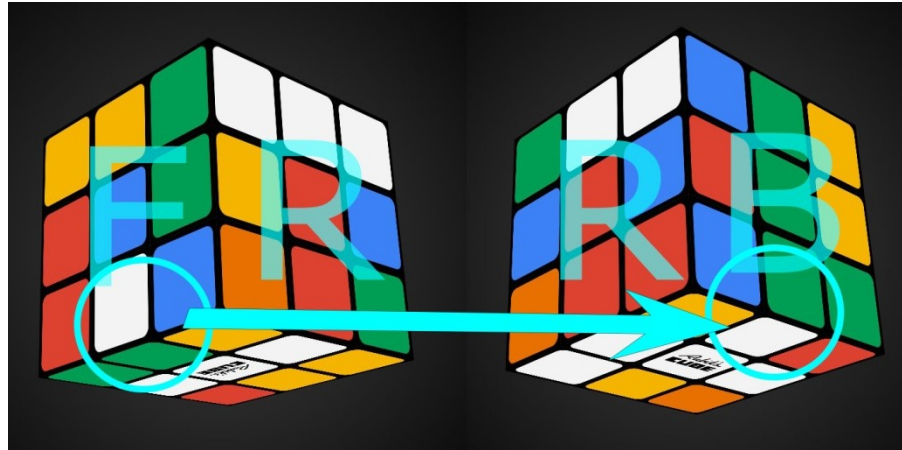
Obr. 43: Presun hrany zo stien FD do LD

Keby nastal prípad, kde je potrebné premiestniť hľadanú hranu na protiľahlú stenu za rovnakých predpokladov, napríklad zo steny 1 (F) na stenu 4 (B) (*Obr. 44*), bolo by to možné rotáciami podľa nasledujúceho úseku kódu:

```

IF cube{1,6} = cube{5,9} THEN
  IF cube{5,2} = cube{4,9} THEN
    F_CW;
    D_CW;
    L_CW;
    D_CCW;
  ENDIF
ENDIF
ENDIF

```



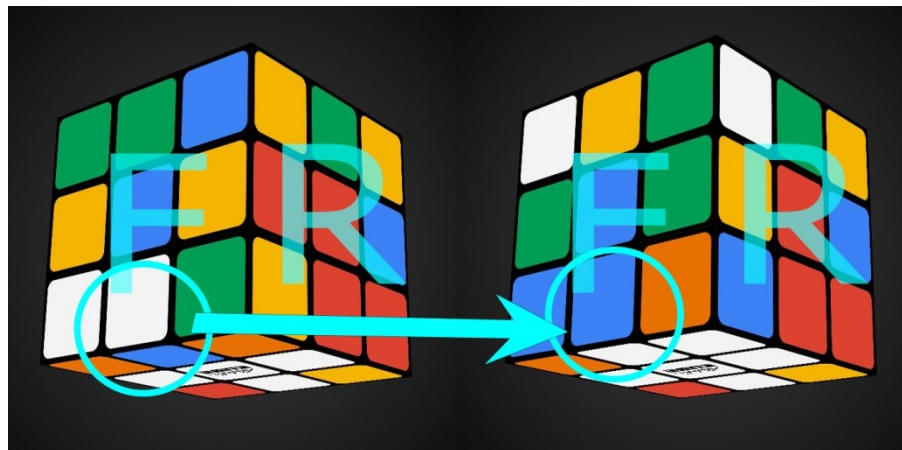
Obr. 44: Presun hrany zo stien FD do BD

Podobne sa postupuje, keď sa hľadaná hrana nachádza v príslušných stenách, ale má prehodené farby (Obr. 45). Vtedy je treba rotovať steny kocky podľa nasledujúceho úseku kódu:

```

IF cube{1,6} = cube{5,9} THEN
  IF cube{5,2} = cube{1,9} THEN
    F_CW;
    D_CCW;
    L_CW;
    D_CW;
  ENDIF
ENDIF
ENDIF

```



Obr. 45: Presun hrany zo stien FD do FD, otočenie orientácie

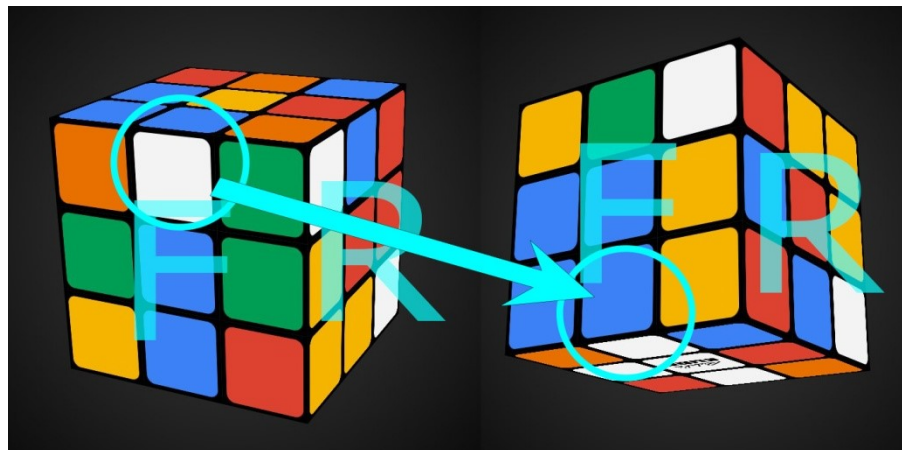
4.3.5 Prípád umiestnenia hľadanej hrany hore

Posledný prípad nastáva, keď sa prvý štvorec hrany nachádza na pozícii 2 (hrana hore), na jednej z možných susedných stien so stenou 5 (D), a jeho druhá časť sa nachádza na stene 2 (U) na jednej z možných pozícií (Obr. 46).

```

IF cube{1,2} = cube{5,9} THEN
  IF cube{2,6} = cube{1,9} THEN
    R_CW;
    U_CCW;
    R_CCW;
    F_CW;
  ELSEIF cube{2,6} = cube{3,9} THEN
    F_CW;
    R_CCW;
    F_CCW;
  ELSEIF cube{2,6} = cube{4,9} THEN
    U_CCW;
    R_CW;
    B_CCW;
    R_CCW;
  ELSEIF cube{2,6} = cube{6,9} THEN
    F_CCW;
    L_CW;
    F_CW;
  ENDIF
ENDIF
ENDIF

```



Obr. 46: Presun hrany zo stien FU do FD, otočenie orientácie

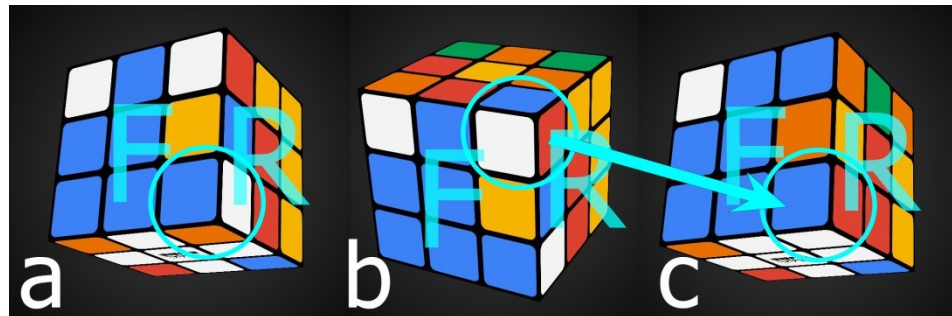
Overovanie podmienok a následné rotácie sú vykonávané na rovnakom princípe, ako to je pri predošlých premiestňovaniach.

4.4 Riešenie prvej vrstvy

Prvá vrstva je riešená na základe začiatocnickej metódy. „Beginner“ metóda bola zvolená z dôvodu autentickosti bežného skladania. Celé to spočíva v sekvencii štyroch základných pohybov. Napríklad je potrebné vyriešiť roh na prednej stene vpravo dole (1,5) (Obr. 47 a). Prvým krokom je hľadanie rohového kúska, ktorý je potrebné na túto pozíciu premiestniť. Keď sa hľadaný roh nachádza niekde v hornej vrstve (nezáleží na otočení), je horná stena otáčaná, až kým roh nezaujme vertikálnu pozíciu nad miestom, kam patrí (Obr.

47 b). Následne sú vykonávané tieto pohyby, pokiaľ nie je na požadovanom mieste a nie je otočený správne (Obr. 47 c). Rotácie sú:

- otočenie pravej steny v smere hodinových ručičiek (R_CW),
- otočenie vrchnej steny v smere hodinových ručičiek (U_CW),
- otočenie pravej steny v protismere hodinových ručičiek (R_CCW),
- otočenie vrchnej steny v protismere hodinových ručičiek (U_CCW).



Obr. 47: Riešenie prvej vrstvy

Keby sa hľadaný roh už nachádzal v spodnej vrstve, bolo by potrebné ho dostať do vrchnej vrstvy. Napríklad, keby bol na pravej stene vpravo dole (3,5), boli by vykonávané už spomíname pohyby, len miesto prednej steny by bolo potrebné rotovať stenu zadnú. Vyzeralo by to takto:

- otočenie zadnej steny v smere hodinových ručičiek (B_CW),
- otočenie vrchnej steny v smere hodinových ručičiek (U_CW),
- otočenie zadnej steny v protismere hodinových ručičiek (B_CCW),
- otočenie vrchnej steny v protismere hodinových ručičiek (U_CCW).

Teraz by sa roh nachádzal hore a pokračovalo by sa podľa prvého postupu.

V kóde ju reprezentuje procedúra pod názvom *first_layer*, ktorá v prvom rade obsahuje testovanie podmienky *WHILE*, pokiaľ všetky rohy nebudú na správnom mieste, orientované správne. V druhom rade obsahuje podmienky *IF*, ktoré hľadajú určité farebné štvorce rohov.

```

IF cube{1,3} = cube{5,9} THEN
  IF cube{2,5} = cube{1,9} AND cube{3,1} = cube{3,9} THEN
    RURU_1;
  ELSEIF cube{2,5} = cube{3,9} AND cube{3,1} = cube{4,9} THEN
    U_CCW;
    BUBU_1;
ENDIF

```

Nad odsekom textu je ukázaná časť kódu, ktorá porovnáva farbu štvorcov hľadaného rohu v stenách 1, 2 a 3 (F, U, R) s hľadanou farbou štvorca v stene 1 (F), patriaceho do steny 5 (D). Pri výskyte odpovedajúceho rohu na tomto mieste by boli vykonávané pohyby podľa kódu nad odsekom textu na základe ostatných dvoch farieb. Napríklad, prvý prípad. Keby sa štvorec zo steny 5 (D) nachádzal na pozícií 3 v stene 1 (F) a jeho susedné štvorce by boli: štvorec zo steny 1 (F) s umiestnením v stene 2 (U) na pozícií 5 a štvorec zo steny 3 (R) s umiestnením v stene 3 (R) na pozícií 1 (*Obr. 47 a*), vykonávala by sa pohybová procedúra s názvom *RURU_1*. Táto procedúra navyše obsahuje testovanie podmienky *WHILE*, dokým nie je roh na svojom mieste, orientovaný farebne správne. Taktiež obsahuje 4 základné pohyby, spomínané o odsek vyššie.

Podobným princípom sú riešené aj ostatné rohy z prvej vrstvy s vyššie popísaným postupom, kde je možno potrebné roh najprv premiestniť na ideálnu pozíciu.

4.5 Riešenie druhej vrstvy

Keďže riešenie druhej vrstvy spočíva v naviazanosti na vrstvu prvú, stále je použitá „Beginner“ metóda. Riešením je umiestňovanie hrán do druhej vrstvy s nutnosťou správnej orientácie a pozície. Nastat' môžu dva prípady. Jednoduchší, keď sa premiestňovaná hrana nachádza v hornej vrstve. Vtedy ju stačí vložiť na správne miesto. Napríklad, vkladaná je hrana, patriaca do steny 1 (F) na pozíciu 4 (hrana vpravo). Spomínanú hranu je potrebné natočiť pomocou otočenia vrchnej steny tak, aby sa nachádzala v stene 6 (L), orientovaná farbou steny 1 (F) na pozícií 2 (hrana hore). Treba vykonať štyri základné pohyby, zhodné s pohybmi v riešení prvej vrstvy pre stenu 3 (R) a po nich ďalšie štyri, ale tentokrát v opačných rotáciách so susednou stenou 1 (F) umiestňovaného kúska. Pohyby sú:

- otočenie pravej steny v smere hodinových ručičiek (*R_CW*),
- otočenie vrchnej steny v smere hodinových ručičiek (*U_CW*),
- otočenie pravej steny v protismere hodinových ručičiek (*R_CCW*),
- otočenie vrchnej steny v protismere hodinových ručičiek (*U_CCW*),
- otočenie prednej steny v protismere hodinových ručičiek (*F_CCW*),
- otočenie vrchnej steny v protismere hodinových ručičiek (*U_CCW*),
- otočenie prednej steny v smere hodinových ručičiek (*R_CW*),
- otočenie vrchnej steny v smere hodinových ručičiek (*U_CW*).

Druhý zložitejší prípad nastane, keď sa premiestňovaná hrana nachádza v druhej vrstve, ale je orientovaná zle alebo je na nesprávnom mieste. Postup je veľmi podobný, len na začiatku je potrebné dostať hrana z nesprávneho uloženia. Napríklad, ak je vkladaná hrana patriaca do steny 1 (F) pozícia 4 (hrana vpravo), ktorá má druhý štvorec na stene 3 (R), orientovaná naopak. Bolo by potrebné vykonať štyri základné pohyby so stenou 3 (R), štyri pohyby v opačných rotáciách so stenou 1 (F), natočenie vrchnej steny pre premiestnenie hrany do steny 6 (L) s orientáciou farby steny 1 (F) na pozícii 2 (hrana hore), štyri základné pohyby so stenou 3 (R) a nakoniec štyri pohyby v opačných rotáciách so stenou 1 (F).

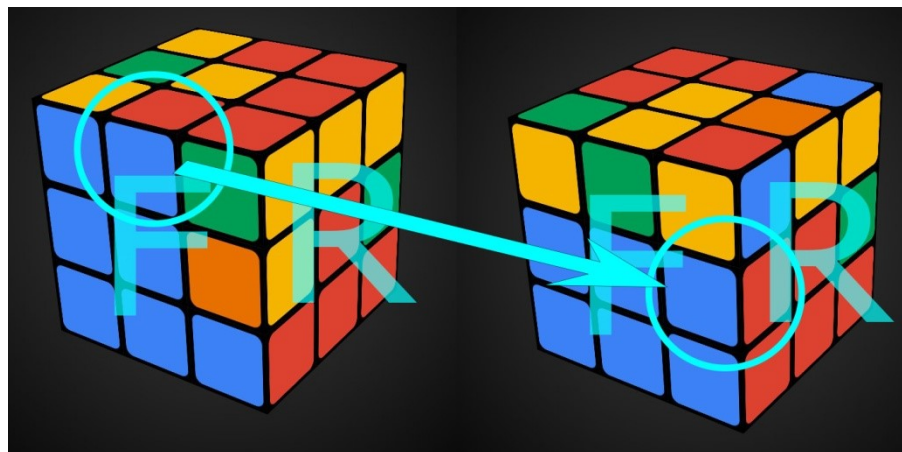
Ako pri riešení prvej vrstvy, tak aj tu bola vytvorená procedúra s názvom *second_layer*, ktorá v prvom rade obsahuje testovanie podmienky *WHILE*, dokým všetky hrany nebudú na správnom mieste, orientované správne. Potom jednotlivé, vyššie spomínané pohyby, sú zapísané nasledovne, kde štvorica základných pohybov je reprezentovaná procedúrou *RURU* a štvorica základných pohybov ale v opačných rotáciách procedúrou *FUFU_R*.

Pre jednoduchší prípad, keď sa premiestňovaná hrana už nachádza v hornej vrstve (Obr. 48):

```

IF cube{1,2} = cube{1,9} THEN
  IF cube{2,6} = cube{3,9} THEN
    U_CW;
    RURU;
    FUFU_R;
  ENDF
ENDIF

```



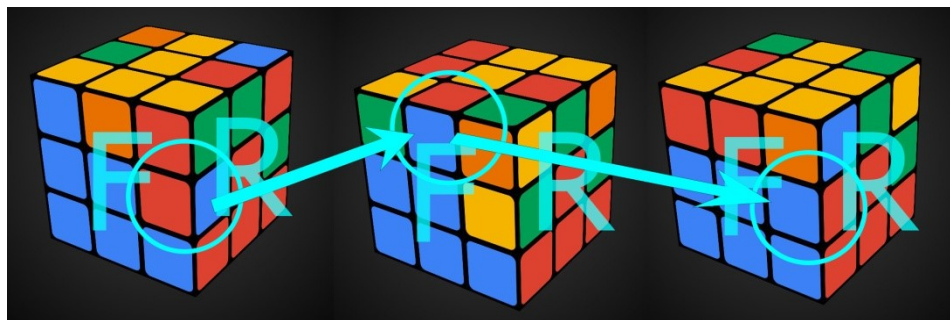
Obr. 48: Riešenie druhej vrstvy – hrana v hornej vrstve

Zložitejší prípad, keď sa premiestňovaná hrana nachádza v druhej vrstve, ale je orientovaná zle alebo je na zlom mieste. V tomto prípade je orientovaná naopak (Obr. 49).

```

IF cube{1,4} = cube{3,9} THEN
  IF cube{3,8} = cube{1,9} THEN
    RURU;
    F_CCW;
    U_CCW;
    F_CW;
    U_CCW;
    RURU;
    FUFU_R;
  ENDIF
ENDIF
ENDIF

```



Obr. 49: Riešenie druhej vrstvy – hrana v druhej vrstve

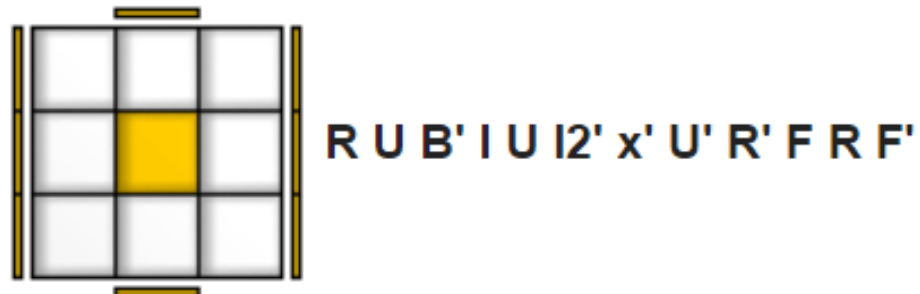
V bloku kódu nad odsekom textu oproti zápisu vyššie je upravená druhá štvorica pohybov, kde miesto štyroch pohybov v opačných rotáciách so stenou 1 (F), sú vykonávané prvé 3 pohyby a tretí zahrňuje otočenie, aby sa hrana nachádzala v stene 6 (L), orientovaná farbou steny 1 (F) na pozícii 2 (hrana hore). S dodržaním vyššie spomínaných pravidiel boli vytvorené pohyby pre všetky hrany patriace do druhej vrstvy.

4.6 Orientácia poslednej vrstvy

Pri orientácii poslednej vrstvy bol algoritmus tvorený na základe pokročilej Fridrichovej metódy, ktorá presúva všetky farby z vrchnej vrstvy umiestnené rôzne do vrchnej steny. Potrebné bolo implementovať 57 čiastkových algoritmov, ktoré boli upravené, aby riešenie nebolo závislé na počiatocnom natočení kocky. To znamená odstránenie pohybov ako **M**, **E**, **S**, **x**, **y**, **z** a **f**, **u**, **r**, **b**, **l**, **d** (vysvetlenie v kapitole 2.1). Úpravy niektorých z týchto pohybov boli vykonané aj z dôvodu potreby viacerých prstov, ktorými však osadený *SmartGripper* nedisponuje. Avšak na základné pohyby, ktorými boli tieto nahradené, stačia dva prsty na každej ruke.

V kóde je táto časť riešenia reprezentovaná procedúrou pod názvom *orient_last_layer*. Na rozdiel od ostatných štádií sa v tomto nenachádza podmienka *WHILE*, z dôvodu pokročilej

techniky skladania, pri ktorej sa vykoná len jedna kombinácia pohybov. Pre tvorenie algoritmov v kóde ako vzor slúžili algoritmy, ktoré sú dostupné na [43]. Žlté štvorce a čiary prezentujú výskyt farebných štvorcov, ktoré patria do steny 2 (U) (Obr. 50).



Obr. 50: Orientácia poslednej vrstvy – Dot 1 [43]

Keď na kocke napríklad vznikne vzor zhodný so vzorom na obrázku, je potrebné aplikovať kombináciu pohybov vypísaných na obrázku. Keďže sa v kombinácii vyskytli pohyby I a x, bolo potrebné algoritmus upraviť do tejto formy:

```

IF cube{1,2} = cube{2,9} AND cube{3,1} = cube{2,9} AND
cube{3,2} = cube{2,9} AND cube{3,3} = cube{2,9} AND
cube{4,2} = cube{2,9} AND cube{6,1} = cube{2,9} AND
cube{6,2} = cube{2,9} AND cube{6,3} = cube{2,9} THEN
  R_CW;
  U_CW;
  B_CCW;
  R_CW;
  B_CW;
  R_CW;
  R_CW;
  U_CCW;
  R_CCW;
  F_CW;
  R_CW;
  F_CCW;
ENDIF

```

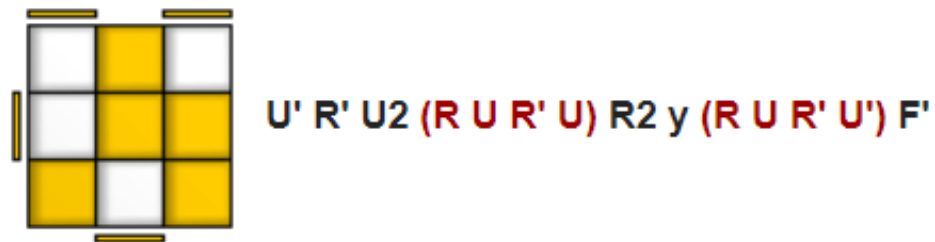
Toto riešenie je platné, keď sa farby zo steny 2 (U) po jednej nachádzajú na pozíciách 2 (hrana hore) v stenách 1 (F) a 4 (B). Keby boli farby zo steny 2 (U) po jednej na pozíciách 2 v stenách 3 (R) a 6 (L), riešenie by sa zmenilo takto:

```

IF cube{1,2} = cube{2,9} AND cube{3,1} = cube{2,9} AND
cube{3,2} = cube{2,9} AND cube{3,3} = cube{2,9} AND
cube{4,2} = cube{2,9} AND cube{6,1} = cube{2,9} AND
cube{6,2} = cube{2,9} AND cube{6,3} = cube{2,9} THEN
  D_CW;
  U_CW;
  R_CCW;
  F_CW;
  R_CW;
  F_CW;
  F_CW;
  U_CCW;
  F_CCW;
  L_CW;
  F_CW;
  L_CCW;
ENDIF

```

Keďže je tento vzor zrkadlovo symetrický, stačia len tieto dve kombinácie pohybov pre vyriešenie orientácie poslednej vrstvy. Pre komplikovanejšie vzory boli vytvorené štyri kombinácie pohybov. Napríklad pre vzor na obrázku (*Obr. 51*).



Obr. 51: Orientácia poslednej vrstvy – Shape |_ 1 [43]

Kód potom vyzerá nasledovne:

```

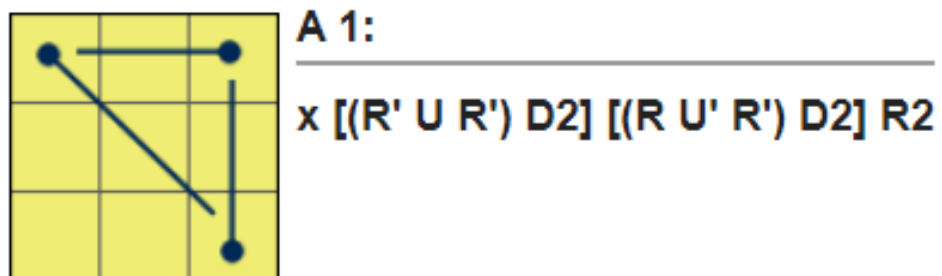
IF cube{1,2} = cube{2,9} AND cube{4,1} = cube{2,9} AND
cube{4,3} = cube{2,9} AND cube{6,2} = cube{2,9} AND
cube{2,2} = cube{2,9} AND cube{2,4} = cube{2,9} AND
cube{2,5} = cube{2,9} AND cube{2,7} = cube{2,9} THEN
  U_CCW;
  R_CCW;
  U_CW;
  U_CW;
  R_CW;
  U_CW;
  R_CCW;
  U_CW;
  R_CW;
  R_CW;
  BUBU;
  R_CCW;
ENDIF

```

4.7 Permutácie rohov poslednej vrstvy

Permutácie poslednej vrstvy sú posledným štádiom skladania kocky. Pri permutáciách sú umiestňované farby z poslednej vrstvy na svoje príslušné miesto. Aj tu, ako pri orientáciách poslednej vrstvy, bol algoritmus tvorený na základe pokročilej Fridrichovej metódy a taktiež niektoré algoritmy museli prejsť úpravou. Dokopy bolo potrebné implementovať 21 čiastkových algoritmov.

V kóde je táto časť riešenia reprezentovaná procedúrou pod názvom *permutate_last_layer*. Na prvom mieste procedúra obsahuje testovanie podmienky *WHILE*, pokiaľ kocka nie je vyriešená. Pri každej neúspešnej otočke cyklu je na konci podmienky *WHILE* rotácia vrchnej vrstvy v smere hodinových ručičiek o 90°. Dôvodom takého riešenia je možná chybnosť natočenia poslednej vrstvy po predošlom štádiu orientácie poslednej vrstvy. Pre tvorenie algoritmov v kóde ako vzor slúžili algoritmy, ktoré sú dostupné na [44]. Na obrázkoch sú bodky, ktoré značia smer permutácie ako hlava šípky (Obr. 52).



Obr. 52: Permutácie rohov poslednej vrstvy – A1 [44]

Napríklad prvý algoritmus, na [44] značený ako A1, je v kóde zapísaný nasledovne:

```

IF cube{1,3} = cube{4,9} AND cube{3,1} = cube{6,9} AND
cube{3,3} = cube{1,9} AND cube{4,1} = cube{3,9} AND
cube{4,3} = cube{3,9} AND cube{6,1} = cube{4,9} AND
cube{1,2} = cube{1,9} AND cube{3,2} = cube{3,9} AND
cube{4,2} = cube{4,9} AND cube{6,2} = cube{6,9} AND
cube{6,3} = cube{6,9} AND cube{1,1} = cube{1,9} THEN
  R_CCW;
  F_CW;
  R_CCW;
  B_CW;
  B_CW;
  R_CW;
  F_CCW;
  R_CCW;
  B_CW;
  B_CW;
  R_CW;
  R_CW;
ENDIF

```

5 IMPLEMENTÁCIA LOGIKY

Táto kapitola je vyhradená komunikácii medzi jednotlivými ramenami, aplikácii získaného riešenia z algoritmu a taktiež pohybmi robota súvisiacimi so skladaním kocky.

5.1 Robtargety

Robtargety sa používajú na definovanie pozície ramena robota a prídavných osí v priestore. Napríklad robtarget pre pozíciu držania kocky ľavého ramena je zapísaný ako konštanta:

```
CONST robtarget pHold_L:=[[300,20,180],[0.49999605,0.49999547,-0.500004835,0.500003639],
[-1,1,-1,4],[140,9E+09,9E+09,9E+09,9E+09,9E+09]];
```

kde prvé 3 čísla určujú polohu nástroja v globálnom súradnicovom systéme (x, y, z) (pokiaľ nie je zadaný iný) vyjadrenú v mm, druhá skupina 4 čísiel určuje orientáciu nástroja vyjadrenú vo forme kvaterniónu ($q1, q2, q3, q4$), tretia skupina 4 čísiel určuje konfiguráciu osí robota ($cf1, cf4, cf6, cfx$). Toto je definované vo forme aktuálnej štvrtinovej otáčky osi 1, osi 4 a osi 6. Prvá kladná štvrtinová otáčka 0 až 90 ° je definovaná ako 0. Význam zložky cfx závisí od typu robota. Posledná a to štvrtá skupina 6 čísiel určuje polohu externých osí ($eax_a, eax_b, eax_c, eax_d, eax_e, eax_f$). Tento číselný reťazec je generovaný automaticky po vytvorení tzv. Targetu v programe a následnom synchronizovaní zo stanice do programovacieho jazyka RAPID.

5.2 Synchronizácia

Keďže robot disponuje dvoma ramenami, je potrebné synchronizovať všetky úkony a pohyby medzi nimi. Každé rameno je prezentované samostatným taskom v programe a to buď T_ROB_L alebo T_ROB_R v závislosti, či ide o rameno ľavé alebo pravé. V hlavom module *MainModule* v oboch spolupracujúcich taskoch sa potom nachádza deklarácia programovej synchronizácie:

```
PERS tasks task_list{2} := [{"T_ROB_L"}, {"T_ROB_R"}];
VAR syncident syncHome;
VAR syncident syncReady;
VAR syncident syncShuffled;
```

kde na prvom mieste sa nachádza definícia systémovej globálnej premennej *PERS* spolu so spolupracujúcimi taskmi. Ďalej musí byť zadeklarovaná identita pre synchronizačný bod *syncident* spolu s názvom použitého bodu. Napríklad ako vytvorené synchronizačné body *syncHome* (synchronizácia pri polohy ramien v domovskej pozícii), *syncReady* (synchroni-

Pohyby robota vykonávané zo získaného riešenia sú riešené v procedúre *solve_the_cube*, ktorá obsahuje testovanie podmienky WHILE, dokým sa nevyskytne prázdny string v poli riešenia *strSolution*.

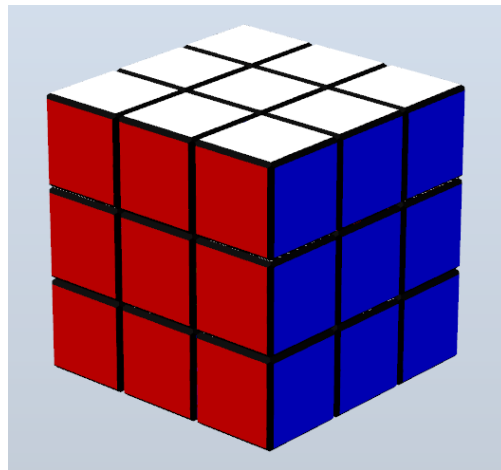
V procedúre *solve_the_cube* sa vyskytujú viaceré pohyby robota v závislosti od toho, či ide o rameno pravé alebo ľavé. Pre rameno ľavé sú to napríklad pohyby: pohyb ramena do polohy držania kocky (*go_to_hold_L*), pohyby ramena pre rotácie jednotlivých stien (*go_to_L*, *go_to_U*, *go_to_D*). Pre rameno pravé sú to obdobné pohyby: pohyb ramena do polohy držania kocky (*go_to_hold_R*), pohyby ramena pre rotácie jednotlivých stien (*go_to_R*, *go_to_F*, *go_to_B*). Inak povedané, ľavé rameno otáča stenami U, L, D a pravé rameno zase stenami R, F a B. Môžu to byť aj rotačné pohyby uchopovača (*SmartGripper*) okolo osi z o $\pm 90^\circ$ v smere alebo protismere hodinových ručičiek *rot_CW* a *rot_CCW*. Signály na ovládanie pripnutia alebo odopnutia kocky k uchopovaču alebo čakanie času potrebného pre vykonanie rotácie kocky.

6 SMART COMPONENT KOCKY

Virtualizovaný model kocky je tvorený na základe Smart Componentov popísaných v kapitole 3.1.3.1. Slúži ako vizuálna predstava reálnej Rubikovej kocky v simulácii, na ktorú sú aplikované pohyby robota získané z algoritmu riešenia.

6.1 Realizácia

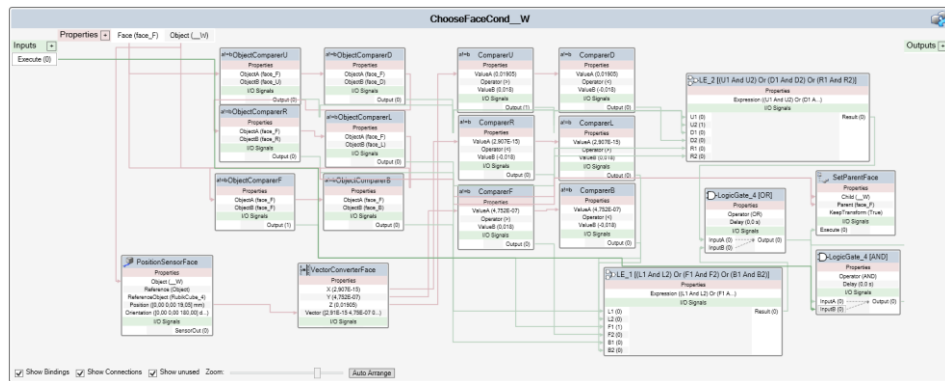
Pri realizácii Smart Componentu kocky bol v prvom rade vytvorený 3D model Rubikovej kocky (Obr. 53), ktorý je tvorený z 27 menších farebných kociek (26 viditeľných kociek a 1 kocka v strede reprezentujúca stredový hriadeľ kvôli jednoduchosti).



Obr. 53: Model Rubikovej kocky

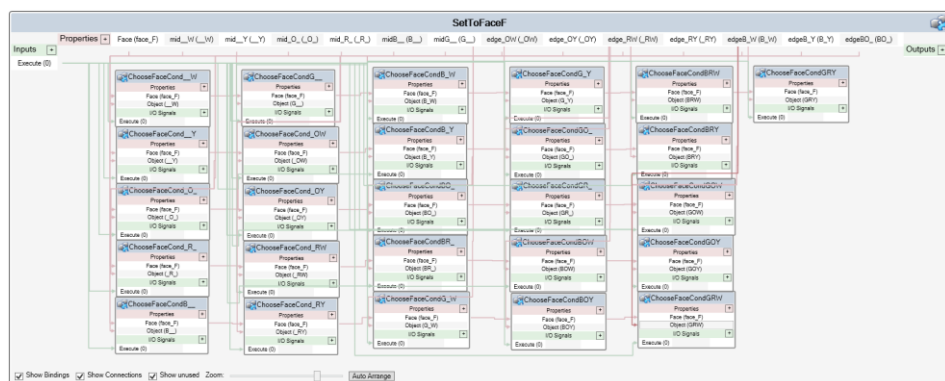
Jednotlivé čiastkové farebné kocky boli pomenované podľa farieb, z ktorých sú zložené. Následne boli tieto čiastkové kocky vložené do tzv. skupiny komponentov (Component Group) pod názvom *rubik_original_DO_NOT_CHANGE* (originál kocky) a vlastnosť tejto kocky bola nastavená na skrytú. Ku skrytej kocke bola vytvorená druhá, už viditeľná, ale totožná kocka a jej čiastkové kocky umiestnené do skupiny *rubik* (kocka, na ktorú sú aplikované pohyby). V skupine *rubik* boli vytvorené vnútorné skupiny komponentov, ktoré predstavujú jednotlivé steny kocky. Obe skupiny boli vložené do jedného Smart Componentu.

V Smart Componente pre každú čiastkovú farebnú kocku bola vytvorená funkcia, ktorá zisťuje, v ktorej stene sa daná kocka aktuálne nachádza (Obr. 54). Funguje to na princípe porovnávania vzdialenosti od stredu súradnicového systému kocky k zvolenému referenčnému bodu, v smere jednej z osí, v závislosti od toho, o ktorú stenu ide.

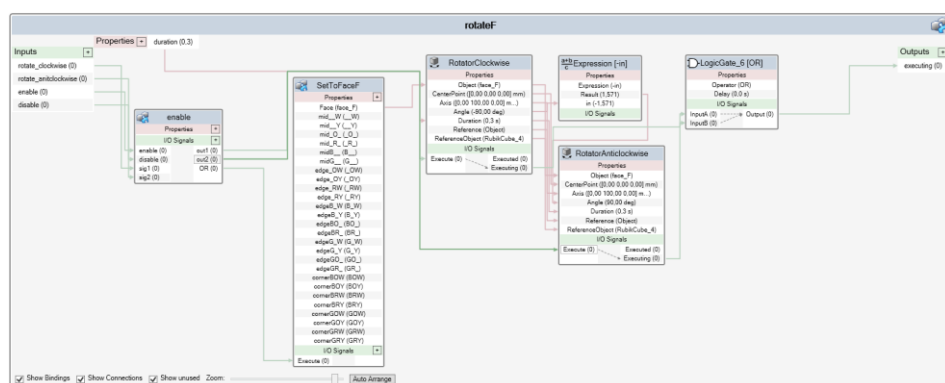


Obr. 54: Realizácia Smart Componentu kocky – ChooseFaceCond__W

Všetky funkcie vytvorené pre každú čiastkovú farebnú kocku boli vložené do funkcie pre každú jednu stenu (Obr. 55).

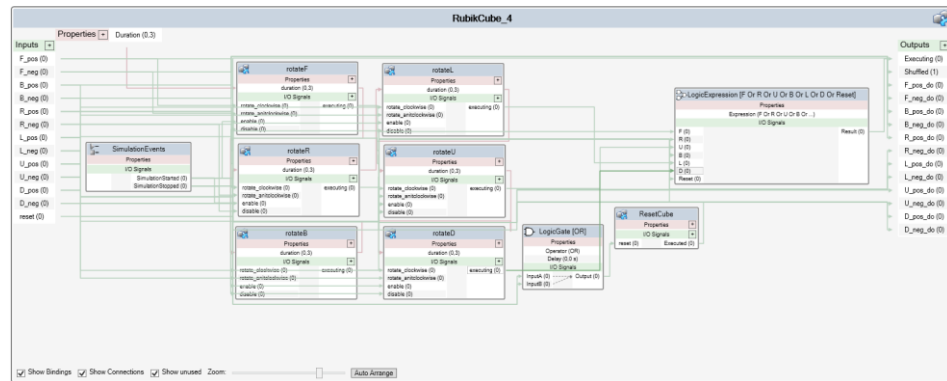


Obr. 55: Realizácia Smart Componentu kocky – SetToFaceF



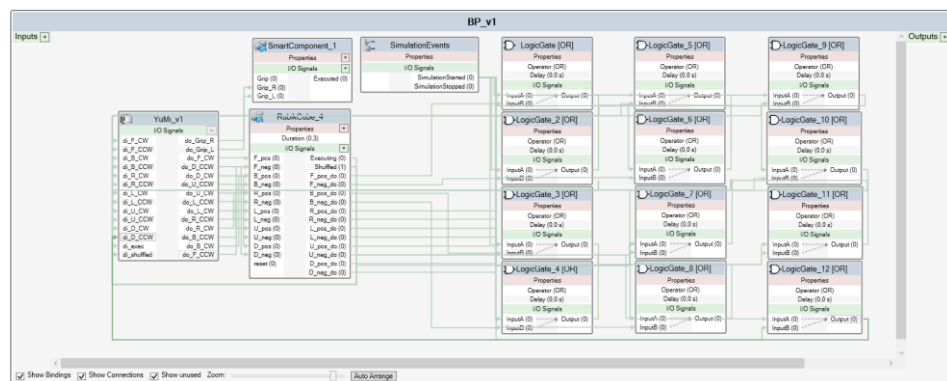
Obr. 56: Realizácia Smart Componentu kocky – rotateF

Vzniknutá funkcia bola potom použitá pre rotácie virtuálnej kocky (Obr. 56), či už v smere hodinových ručičiek alebo v protismere (Obr. 57).



Obr. 57: Realizácia Smart Componentu kocky – smer rotácie

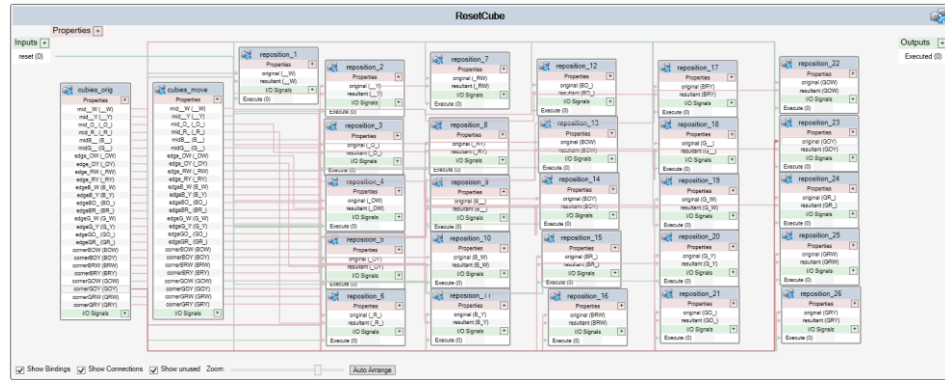
O tom, ktorá strana a v akom smere bude rotovaná, rozhodujú vstupné signály privádzané z kontroléra na Smart Component alebo taktiež zo vstupov užívateľského rozhrania na zadanie počiatočného zamiešania (Obr. 58).



Obr. 58: Realizácia Smart Componentu kocky – vstupné signály

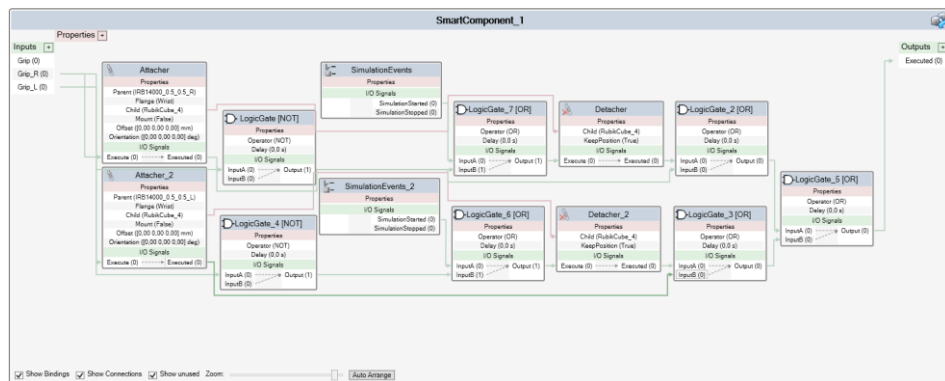
Pre zhrnutie: zo získaného vstupného signálu, ktorý určuje stenu rotácie a jej smer, Smart Component naplní správnu stenovú skupinu komponentu čiastkovými farebnými kockami podľa výskytu v aktuálnej stene a následne vykoná rotáciu tejto skupiny v preddefinovanom smere. Pri ďalšej rotácii, napríklad susednej steny, vyberie potrebné kocky z pôvodnej skupiny a vloží do novej. Pokračuje klasicky.

Smart Component kocky disponuje aj funkciou *reset*, ktorá vráti kocku do pôvodného zloženého stavu (Obr. 59).



Obr. 59: Realizácia Smart Componentu kocky – ResetCube

Nepriamou súčasťou Smart Componentu kocky je aj Smart Component, ktorý sa stará o pripnutie a odopnutie kocky k uchopovaču pri rotáciách stien D a B, kde je nutné kocku vychýliť zo stálej rovnovážnej polohy, aby ramena robota neprechádzali cez telo robota, stôl alebo inú podložku (Obr. 60).

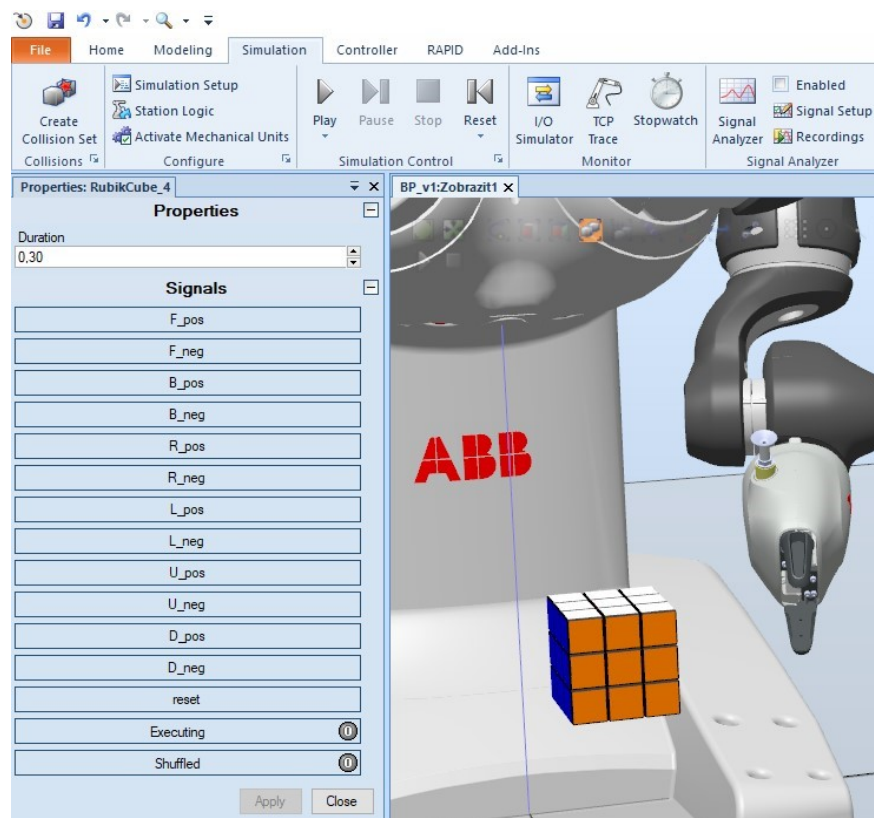


Obr. 60: Realizácia Smart Componentu kocky – Attacher/Detacher

7 SIMULÁCIA A VÝSLEDKY

7.1 Ovládanie

Po spustení simulácie program čaká na zadanie počiatočného zamiešania kocky pomocou pripravených tlačidiel, ktoré je možné zobraziť po kliknutí pravým tlačidlom myši na vytvorený Smart Component kocky a následnom vybratí položky *Properties* z kontextového okna. Po otvorení *Properties* sa zobrazí okno na ľavej strane programového prostredia, ktoré obsahuje všetky možné pohyby spolu s tlačidlami *reset*, *Executing* a *Shuffled* (Obr. 61). Pohybové tlačidlá pre určité steny sú zapísané napríklad ako *F_pos* a *F_neg* v závislosti od smeru požadovanej rotácie (*pos* – pre smer rotácie v smere hodinových ručičiek, *neg* – pre smer rotácie v protismere hodinových ručičiek). Tlačidlo *reset*, ktoré slúži na vrátenie kocky do pôvodného (nezamiešaného) stavu a jeho funkcionality funguje len pri vypnutej simulácii. Tlačidlo *Executing*, ktoré slúži na identifikáciu pri rotovaní kocky, kde stav 1 znamená, že kocka aktuálne vykonáva pohyb rotácie, a stav 0 kedy kocka nevykonáva pohyb rotácie. Nakoniec tlačidlo *Shuffled*, ktoré slúži na spustenie simulácie riešenia kocky po zamiešaní podľa získaného reťazca pohybov zo sekcie algoritmu.



Obr. 61: Ovládanie inteligentného komponentu kocky

7.2 Priemerný počet pohybov

Priemerný počet rotácií, počas simulácie riešenia Rubikovej kocky, je 117 za priemerný čas 655 sekúnd (10 min 55 s). Priemerný počet rotácií a priemerný čas celkovej simulácie bol získaný ako priemer z 25-tich simulácií, kde každá simulácia riešila rôzne riešenie. Rubikova kocka bola zamiešaná náhodne generovanými 15-imi pohybmi, kde dvojice rovnakých pohybov o 90° (pohyb o 180°) boli rátané ako jeden. Najvyšší počet rotácií bol 143 za priemerný čas 828 sekúnd (13 min 48 s). Na druhej strane, najnižší počet rotácií bol 95 za priemerný čas 519 sekúnd (8 min 39 s).

7.3 Prípadné vylepšenia

Za prípadné vylepšenia by sa dalo považovať prídanie priemyselnej kamery a vytvorenie tak celku, kde by bolo možné kocku zamiešať fyzicky a robot by si kocku načítal pomocou kamery, vyhodnotil by pozície jednotlivých farebných štvorcov, preniesol dané farby do reťazca zamiešania kocky a následne kocku vyriešil. Ďalším vylepšením by mohlo byť prídanie rýchlejšieho algoritmu (Thistlethwaite 45), kde by mohlo vzniknúť zaujímavé porovnanie výsledkov v efektívnosti riešení.

ZÁVER

Táto práca ukazuje použitie kolaboratívneho dvojramenného robota od firmy ABB s označením IRB 14000 YuMi pri riešení Rubikovej kocky za použitia naprogramovaného algoritmu v podobnosti s riešením ľudským, navrhnutého v simulačnom prostredí RobotStudio. Novovytváraná učebňa v priestoroch fakulty ovplyvnila výber použitého robota, ktorý by mohol slúžiť ako predvážiaci model pri rôznych podujatiach po využití výsledkov práce k reálnemu systému.

Vytvorený model Rubikovej kocky v simulačnom prostredí funguje podľa očakávania. Vykonávané rotácie sú presné. Princíp fungovania modelu kocky je postavený na Smart Componentoch, ktoré sú súčasťou prostredia RobotStudio. Jeho riešenie nadobúda komplexný charakter, čoho dôkazom je rozsiahla štruktúra.

Vytvorený algoritmus je zmesou dvoch existujúcich metód riešenia, a to metódy začiatocníkovej a metódy pokročilej, označovanej taktiež CFOP. Niektoré pohyby z použitých metód boli pozmenené tak, aby ich bolo možné vykonávať robotom, ktorý má k dispozícii len dva prsty na koncovom efektore. Počas fázy testovania vytvorený algoritmus nevykazoval žiadne nedostatky ani chyby v čase tvorenia riešenia. Funkčnosť algoritmu pre každú kombináciu zamiešanej kocky sa však zaručiť nedá, pretože otestovanie 43 triliónov kombinácií by bolo časovo náročné a nie je súčasťou tejto práce. Pohyby robotických ramien boli odladené tak, aby ramená neprechádzali jedno cez druhé a ani cez telo robota, či stôl alebo inú podložku.

Súčasťou práce je vytvorená teoretická časť, ktorá potenciálnemu čitateľovi poskytuje podrobnú analýzu a históriu Rubikovej kocky, ale aj pár slov o vynálezcovi samom. Spomenutá je aj firma ABB spoločne s robotom, ktorý v práci zastupuje dôležitú úlohu. Celý algoritmus bol implementovaný priamo v programovacom jazyku robota RAPID a nespolieha sa tak na externé prvky a knižnice.

ZOZNAM POUŽITEJ LITERATÚRY

- [1] TASR. Vynálezca najslávnejšej farebnej kocky Ernő Rubik dnes jubiluje. *Teraz.sk* [online]. 2014 [cit. 2020-03-03]. Dostupné z: <https://www.teraz.sk/magazin/vynalezca-najslavnejsej-farebnej-kock/91269-clanok.html>
- [2] Otec vynálezcu Rubikovej kocky Ernő Rubik pochádzal z Piešťan, od jeho úmrtia dnes uplynulo dvadsať rokov. *PNky.sk* [online]. 2017 [cit. 2020-03-06]. Dostupné z: <https://www.pnky.sk/koktail/otec-vynalezcu-rubikovej-kocky-ktory-sa-narodil-a-zil-v-piestanoch-zomrel-pred-dvadsiatimi-rokmi/>
- [3] Ernő Rubik [online]. 2020 [cit. 2020-03-08]. Dostupné z: https://en.wikipedia.org/wiki/Ern%C5%91_Rubik
- [4] Our heritage [online]. 2019 [cit. 2020-03-11]. Dostupné z: <https://www.rubiks.com/en-us/about>
- [5] Rubik's Cube [online]. 2020 [cit. 2020-03-11]. Dostupné z: https://en.wikipedia.org/wiki/Rubik%27s_Cube
- [6] Important Dates in Rubik's Cube History. © *Ruwix* [online]. 2018 [cit. 2020-03-12]. Dostupné z: <https://ruwix.com/the-rubiks-cube/history-rubiks-cube/important-dates-timeline/>
- [7] ZENG, Da-Xing, Ming LI, Juan-Juan WANG, Yu-Lei HOU, Wen-Juan LU a Zhen HUANG. Overview of Rubik's Cube and Reflections on Its Application in Mechanism. *Chinese Journal of Mechanical Engineering* [online]. 2018 [cit. 2020-03-15]. Dostupné z: <https://cjme.springeropen.com/articles/10.1186/s10033-018-0269-7>
- [8] WATCH: Here's How You Solve a 17x17x17 Rubik's Cube [online]. 2015 [cit. 2020-03-20]. Dostupné z: <https://www.sciencealert.com/watch-here-s-how-you-solve-a-17x17x17-rubiks-cube>
- [9] 33x33x33 Rubik's Cube - World Record [online]. 2017 [cit. 2020-03-23]. Dostupné z: <https://ruwix.com/blog/33x33x33-rubiks-cube/>
- [10] The Mathematics of the Rubik's Cube: Introduction to Group Theory and Permutation Puzzles [online]. 2009 [cit. 2020-03-26]. Dostupné z: <http://web.mit.edu/sp.268/www/rubik.pdf>

- [11] Rubikova kostka [online]. 2020 [cit. 2020-03-27]. Dostupné z: https://cs.wikipedia.org/wiki/Rubikova_kostka
- [12] SINGMASTER, David. *Rubik's magic cube*. London, England: Enslow Publishers, 1981. ISBN 0-89490-043-9
- [13] BERLEKAMP, Elwyn R., John H. CONWAY a Richard K. GUY. *Winning Ways for your mathematical plays*. Natic, Massachusetts: A K Peters, 2003. ISBN 1-56881-130-6.
- [14] BOSSERT, Patrick. *You Can Do the Cube*. Puffin original, 1981. ISBN 0-14031-483-0
- [15] KORF, Richard E. *Learning to solve problems by searching for macro-operators*. Massachusetts: Pitman Publishing, 1985. ISBN 978-0-273-08690-1
- [16] Advanced Rubik's Cube notation © *Ruwix* [online]. 2016 [cit. 2020-04-02]. Dostupné z: <https://ruwix.com/the-rubiks-cube/notation/advanced/>
- [17] Rubik's Cube Move Notation [online]. 2020 [cit. 2020-04-05]. Dostupné z: <https://jperm.net/3x3/moves>
- [18] How to solve the Rubik's Cube? © *Ruwix* [online]. 2016 [cit. 2020-04-05]. Dostupné z: <https://ruwix.com/the-rubiks-cube/how-to-solve-the-rubiks-cube-beginners-method/>
- [19] CFOP method [online]. 2020 [cit. 2020-04-12]. Dostupné z: https://www.speedsolving.com/wiki/index.php/CFOP_method
- [20] Jessica Fridrich [online]. 2020 [cit. 2020-04-12]. Dostupné z: https://en.wikipedia.org/wiki/Jessica_Fridrich
- [21] Rubik's Cube solution with advanced Fridrich (CFOP) method © *Ruwix* [online]. 2017 [cit. 2020-04-12]. Dostupné z: <https://ruwix.com/the-rubiks-cube/advanced-cfop-fridrich/>
- [22] Roux method [online]. 2020 [cit. 2020-04-16]. Dostupné z: https://www.speedsolving.com/wiki/index.php/Roux_method
- [23] Roux method [online]. [cit. 2020-04-16]. Dostupné z: https://rubiks.fandom.com/wiki/Roux_Method
- [24] ZZ method [online]. 2020 [cit. 2020-04-21]. Dostupné z: https://www.speedsolving.com/wiki/index.php/ZZ_method

- [25] Different Rubik's Cube Solving Methods © *Ruwix* [online]. 2017 [cit. 2020-04-21]. Dostupné z: <https://ruwix.com/the-rubiks-cube/different-rubiks-cube-solving-methods/>
- [26] HEIT, Joren. *Rubik's Cube Simulator and Solver* [online]. 2011 [cit. 2020-04-26]. Dostupné z: <https://www.mathworks.com/matlabcentral/fileexchange/31672-rubik-s-cube-simulator-and-solver>
- [27] Morwen Thistlethwaite [online]. 2018 [cit. 2020-04-26]. Dostupné z: https://www.speedsolving.com/wiki/index.php/Morwen_Thistlethwaite
- [28] Thistlethwaite's 52-move algorithm [online]. [cit. 2020-04-26]. Dostupné z: <https://www.jaapsch.net/puzzles/thistle.htm>
- [29] God's Number © *Ruwix* [online]. 2017 [cit. 2020-04-28]. Dostupné z: <https://ruwix.com/the-rubiks-cube/gods-number/>
- [30] ROKICKI, Tomas, Herbert KOCIEMBA, Morley DAVIDSON, John DETHRIDGE a Lucas GARRON. *God's Number is 20* [online]. Seven Towns [cit. 2020-05-03]. Dostupné z: <https://www.cube20.org/>
- [31] Kto sme - ABB v skratke [online]. 2020 [cit. 2020-05-09]. Dostupné z: <https://new.abb.com/sk/o-nas/abb-v-skratke>
- [32] History of ABB [online]. 2020 [cit. 2020-05-09]. Dostupné z: <https://global.abb/group/en/about/history>
- [33] ABB unveils the future of human-robot collaboration: YuMi® [online]. Zurich, Switzerland, 2014 [cit. 2020-05-09]. Dostupné z: <https://new.abb.com/news/detail/13110/abb-unveils-the-future-of-human-robot-collaboration-yumir>
- [34] YuMi® - vytvorme spolu budúcnosť automatizácie [online]. 2020 [cit. 2020-05-09]. Dostupné z: <https://new.abb.com/products/robotics/sk/roboty/yumi>
- [35] YuMi® creating an automated future together. 9AKK106354A3254. ABB. 2019
- [36] Product manual - IRB 14000 gripper. 3HAC054949-001. ABB. 2015-2018
- [37] Application manual – Integrated Vision. 3HAC044251-001. ABB, 2018.
- [38] Komunikace přes rozhraní OPC [online]. [cit. 2020-05-24]. Dostupné z: <https://www.promotic.eu/cz/pmdoc/Subsystems/Comm/OPC/OPC.htm>

- [39] Download RobotStudio with RobotWare and PowerPacs [online]. 2020 [cit. 2020-05-26]. Dostupné z: <https://new.abb.com/products/robotics/robotstudio/downloads>
- [40] RAPID [online]. 2020 [cit. 2020-05-30]. Dostupné z: <https://en.wikipedia.org/wiki/RAPID>
- [41] SHAW, Keith a Josh FRUHLINGER. What is a digital twin and why it's important to IoT. *Network World* [online]. 2019 [cit. 2020-06-10]. Dostupné z: <https://www.networkworld.com/article/3280225/what-is-digital-twin-technology-and-why-it-matters.html>
- [42] What is a digital twin and why it's important to IoT [online]. 2020 [cit. 2020-06-11]. Dostupné z: <https://www.plm.automation.siemens.com/global/en/our-story/glossary/digital-twin/24465>
- [43] Step 3 - Orient last layer - OLL [online]. 2016 [cit. 2020-06-24]. Dostupné z: <https://ruwix.com/the-rubiks-cube/advanced-cfop-fridrich/orient-the-last-layer-oll/>
- [44] Step 4 - Permutate the last layer - PLL [online]. 2016 [cit. 2020-06-29]. Dostupné z: <https://ruwix.com/the-rubiks-cube/advanced-cfop-fridrich/permutate-the-last-layer-pll/>

ZOZNAM POUŽITÝCH SYMBOLOV A SKRATIEK

g	Gram
kg	Kilogram
km	Kilometer
m	Meter
m/s	Rýchlosť v metroch za sekundu
m/s ²	Zrýchlenie v metroch za sekundu
mm	Milimeter
OPC	Open Platform Communications
OPC-UA	Open Platform Communications - Unified Architecture
TCP/IP	Transmission Control Protocol/Internet Protocol

ZOZNAM OBRÁZKOV

Obr. 1: Enró Rubik [1].....	11
Obr. 2: Vetroň R-26 Góbé [2].....	12
Obr. 3: Patent Erna Rubika – Magická kocka ako puzzle [4].....	13
Obr. 4: Space Invader – Výstava pouličného umelca [4]	14
Obr. 5: The Snake a hlavolam Magic [4].....	15
Obr. 6: Jiugong mapa [7]	15
Obr. 7: Trojrozmerný magický štvorec a dvojradová kocka [7].....	16
Obr. 8: Stará verzia Rubikovej kocky [7].....	17
Obr. 9: Tri druhy drážky a čapu – spojenia Rubikovej kocky [7]	18
Obr. 10: Rubikova kocka [7]	18
Obr. 11: Diely Rubikovej kocky [7]	18
Obr. 12: Časti centrálného dielu [7].....	19
Obr. 13: Spojenia v Rubikovej kocke [7]	19
Obr. 14: Režimy kontaktu [7]	20
Obr. 15: Vnútorne časti Rubikovej kocky [7].....	20
Obr. 16: Lubanský zámok po zložení – lopta [7]	21
Obr. 17: Časti Lubanského zámku – lopta [7].....	21
Obr. 18: Lubanské zámky [7]	21
Obr. 19: Druhy Rubikovej kocky – kubické [7]	23
Obr. 20: Druhy Rubikovej kocky – špeciálne [7].....	23
Obr. 21: Rotácie kocky [17]	26
Obr. 22: Biely kríž	27
Obr. 23: L v ľavom zadnom rohu	28
Obr. 24: Jessica Fridrich [20]	29
Obr. 25: Jednotlivé kroky CFOP [21].....	29
Obr. 26: Blok 1x2x3 [23].....	30
Obr. 27: Roux – krok 3 a 5 [23].....	31
Obr. 28: Pomer medzi počtom pohybov a počtom pozícií [29].....	33
Obr. 29: The Super-Flip.....	33
Obr. 30: IRB 14000 YuMi [34]	35
Obr. 31: Pracovný priestor robota IRB 14000 YuMi [35].....	36
Obr. 32: SmartGripper [36]	36

Obr. 33: Kontextové okno komponentov	38
Obr. 34: Použitie Smart Componentov v praxi.....	38
Obr. 35: Reálna kocka olepená číslami pre lepšiu vizualizáciu	42
Obr. 36: Rozložená kocka s použitým označením.....	42
Obr. 37: Presun hrany zo stien FD do RD, orientovanej správne.....	46
Obr. 38: Presun hrany zo stien FU do FD, orientovanej správne	47
Obr. 39: Presun hrany zo stien FR do RD	48
Obr. 40: Presun hrany zo stien FR do FD.....	48
Obr. 41: Presun hrany zo stien FR do LD.....	49
Obr. 42: Presun hrany zo stien FD do RD	50
Obr. 43: Presun hrany zo stien FD do LD	50
Obr. 44: Presun hrany zo stien FD do BD	51
Obr. 45: Presun hrany zo stien FD do FD, otočenie orientácie	51
Obr. 46: Presun hrany zo stien FU do FD, otočenie orientácie	52
Obr. 47: Riešenie prvej vrstvy	53
Obr. 48: Riešenie druhej vrstvy – hrana v hornej vrstve	55
Obr. 49: Riešenie druhej vrstvy – hrana v druhej vrstve	56
Obr. 50: Orientácia poslednej vrstvy – Dot 1 [43]	57
Obr. 51: Orientácia poslednej vrstvy – Shape _ 1 [43].....	58
Obr. 52: Permutácie rohov poslednej vrstvy – A1 [44].....	59
Obr. 53: Model Rubikovej kocky	63
Obr. 54: Realizácia Smart Componentu kocky – ChooseFaceCond__W	64
Obr. 55: Realizácia Smart Componentu kocky – SetToFaceF	64
Obr. 56: Realizácia Smart Componentu kocky – rotateF	64
Obr. 57: Realizácia Smart Componentu kocky – smer rotácie.....	65
Obr. 58: Realizácia Smart Componentu kocky – vstupné signály	65
Obr. 59: Realizácia Smart Componentu kocky – ResetCube	66
Obr. 60: Realizácia Smart Componentu kocky – Attacher/Detacher	66
Obr. 61: Ovládanie inteligentného komponentu kocky	67

ZOZNAM PRÍLOH

P I Obsah CD

PRÍLOHA P I: OBSAH CD

Štruktúra obsahu priloženého CD:

- Adresár **Text bakalárskej práce** – obsahuje text bakalárskej práce vo formáte PDF/A.
- Adresár **Riešenie** – obsahuje program s riešením Rubikovej kocky v simulačnom prostredí RobotStudio