

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY



Softvérová podpora vyučovania matematiky Hejného metódou -
prostredie Siete telies

BAKALÁRSKA PRÁCA

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

**Softvérová podpora vyučovania matematiky Hejného metódou -
prostredie Siete telies**

BAKALÁRSKA PRÁCA

Študijný program: Aplikovaná informatika
Študijný odbor: Aplikovaná informatika
Školiace pracovisko: Katedra aplikovanej informatiky
Vedúci práce: RNDr. Peter Borovanský, PhD.
Konzultant: RNDr. Dagmar Môťovská, PhD.



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Jana Oravcová
Študijný program: aplikovaná informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Softvérová podpora vyučovania matematiky Hejného metódou - prostredie Siete telies
Educational software for Hejny's method of mathematics teaching - environment Cube Nets

Anotácia: Aplikácie sa opierajú o didaktickú kvalitu vyučovania matematiky Hejného metódou, zadania úloh budú vyberané z učebníc matematiky Hejný, M., Jirotková, D., Slezáková, J., Bomeroová, E., Michnová, J.: MATEMATIKA 1.-5., učebnice pro základní školy, Fraus, 2007-2011. Samotné matematické prostredia z týchto učebníc poskytujú gradáciu, flexibilitu a počítajú s interaktivitou, čo sa týka stvárnenia učebnej látky, úloh na riešenie, aj stratégií riešenia. Tieto vlastnosti prostredí treba využiť a preniesť do navrhovaného softvéru. Navrhovaný softvér ponúkne jednotlivým žiakom dostatočné množstvo úloh na jednotlivých úrovniach, podľa ich individuálnych potrieb, čím bude prínosom pre vyučovanie Hejného metódou. Zároveň treba zabezpečiť technickú kvalitu softvéru, kvalitu grafiky, používateľský komfort, prehľadnosť, spoľahlivosť a rýchlosť.

Cieľ: Cieľom práce je vytvoriť mobilnú aplikáciu (pre tablet) na tému zvoleného prostredia Hejného matematiky (HM). Aplikácia pre prvý stupeň ZŠ musí spĺňať zásady tvorby didaktického softvéru. Aplikácia musí byť testovaná na skupine žiakov, a následne upravená podľa zistených potrieb a event. nedostatkov. Zvolené prostredie HM pokrýva viacero typovo odlišných graduujúcich úloh/úrovní zodpovedajúcich konceptom, ktoré žiaci na danej úrovni objavujú. Aplikácia precvičuje každú úlohu/úroveň na sade predvolených a generovaných zadaní. Až po jej zvládnutí môže žiak pokročiť do ďalšej úrovne. Žiak má možnosť vytvoriť vlastné zadanie v rámci každej úlohy/úrovne. Pri návrhu nového zadania (ako aj pri jeho riešení) aplikácia indikuje počet existujúcich/zostávajúcich riešení daného zadania. Generátor zadaní musí generovať zadania s rozumným počtom existujúcich riešení. Aplikácia si ukladá výsledky práce žiaka, ponúka možnosť priebežnej kontroly a prehľad hodnotenia úspešnosti. Prvé testovanie s deťmi v triede sa predpokladá v apríli, druhé testovanie v triede sa predpokladá v júni.

Vedúci: RNDr. Peter Borovanský, PhD.
Konzultant: RNDr. Dagmar Môtovská, PhD.
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: prof. Ing. Igor Farkaš, Dr.



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

Dátum zadania: 23.06.2019

Dátum schválenia: 14.10.2019

doc. RNDr. Damas Gruska, PhD.
garant študijného programu

študent

vedúci práce

PodĎakovanie V prvom rade sa chcem poďakovať vedúcemu tejto bakalárskej práce, RNDr. Petrovi Borovanskému, PhD. za ochotu, nápady, pripomienky a korekciu. V druhom rade ďakujem konzultantke práce RNDr. Dagmar Môťovskej, PhD. za umožnenie prítomnosti na vzorovej hodine matematiky s postupmi Hejného metódy na 1. súkromnom gymnáziu, Bajkalská 10, Bratislava a za jej cenné rady pri tvorbe aplikácie. Ďakujem svojej rodine za preukázané pochopenie a trpezlivosť a svojmu priateľovi Martinovi, jeho bratovi Jakubovi a svojej sesternici Lucii za ochotu podieľať sa na testovaní aplikácie.

Abstrakt v štátnom jazyku

Kľúčové slová:

Abstract

Keywords:

Obsah

1	Teoretické východiská	10
1.1	Edukačný softvér	10
1.2	Hejného metóda	11
1.2.1	Vzorová hodina vedená Hejného metódou	14
1.3	Využitie Hejného metódy v tvorbe edukačného softvéru	15
1.3.1	Prostredie Biland	16
1.3.2	Prostredie Parkety	17
1.4	Prostredie Siete kocky	18
1.4.1	Sieť kocky	19
1.5	Podobné riešenia	20
1.5.1	Nets of 3D Shapes - Geometry Game	21
1.5.2	Hrací kostka	21
1.6	Technológie	22
2	Návrh	24
2.1	Logika gradácie úrovní	24
2.2	Grafické užívateľské rozhranie	25
2.2.1	Hlavné menu	25
2.2.2	Prvá úroveň	27
2.2.3	Druhá úroveň	29
2.2.4	Editor prvej a druhej úrovne	30
2.2.5	Tretia úroveň	31
2.2.6	Štvrtá úroveň	32
2.2.7	Editor tretej a štvrtej úrovne	33
3	Implementácia	35
3.1	Logika generovanie úloh	35
3.2	Vyhodnocovanie úloh	36
3.2.1	Reprezentácia objektov	37
3.2.2	Algoritmus	38

1 Teoretické východiská

V tejto kapitole sa zoznámime so základnými aspektami tvorby edukačného softvéru, s princípmi, ktoré je možné použiť, ako aj s požiadavkami na výsledný produkt. Bližšie si predstavíme Hejného metódu ako nástroj vo výuke na základných školách, vysvetlíme jej hlavné princípy a ukážeme konkrétne príklady využitia tejto metódy v tvorbe edukačného softvéru.

1.1 Edukačný softvér

Softvér môžeme edukačným nazvať práve vtedy, keď je cieľavedome používaný na podporu učenia a učenia sa. Zhrnieme si, aké základné vlastnosti, ktoré by mal spĺňať [1]:

- mal by plniť vytýčený vzdelávací cieľ
- musí byť ľahko ovládateľný
- má mať príjemné užívateľské rozhranie
- má upútať žiaka
- mal by pôsobiť na čo najviac zmyslov žiaka
- má byť primeraný žiakovmu vnímaniu
- má osloviť žiaka hravou a pútavou formou

V dnešnej dobe je vďaka technologickému rozmachu umožnené väčšine školám používať mnohé technológie v rámci výuky, čím sa naskytá potreba pre tvorbu takéhoto softvéru. Je veľmi podstatné uvedomiť si význam takéhoto softvéru v danej cieľovej skupine a tvorcovia teda musia prikladať veľký dôraz na dodržanie špecifických kvalitatívnych požiadaviek [2]:

- Edukačný softvér má byť stabilný a poskytovať používateľom vysoký technický, technologický a používateľský komfort.
- Prezentovaný digitálny obsah má dôkladne, metodicky správne, kvalitne

- Ovládanie edukačného softvéru má byť jednoduché a intuitívne. Používateľské grafické rozhranie musí byť prehľadné a zvolené nástroje na ovládanie softvéru zrozumiteľné.
- Digitálny obsah má byť otvorený, ponúkať rôzne interaktívne edukačné aktivity podporujúce učenie a poznávanie, poskytovať spätnú väzbu a v maximálnej možnej miere využívať vizualizáciu spolu s multimédiami.
- Edukačný softvér musia dopĺňať rôzne podporné didaktické materiály, pracovné listy, metodické príručky ako s jeho digitálnym obsahom správne pracovať a zmysluplne ho používať na vyučovaní, a podrobne mapovať danú problematiku, rešpektovať učebné osnovy a využívať najmodernejšie technológie spracovania informácií.

1.2 Hejného metóda

Táto učebná metóda bola publikovaná českým matematikom Milanom Hejným na podnet nespokojnosti so spôsobom výučby matematiky na základných školách. Dnes sa táto metóda využíva na školách vo viacerých krajinách a sú s ňou zoznámení aj študenti programu Učiteľstvo pre 1. stupeň pedagogických fakúlt Univerzity Karlovej v Prahe a Ostravskej univerzity v Ostrave [3]. Jej hlavným rozdielom od klasického prístupu výuky matematiky je, že deti riešia úlohy samostatne, formou experimentov a diskusie. Poznatky si tak osvojujú prirodzenejšou cestou a natrvalo. Učiteľ tu nemá za úlohu vysvetľovať ani opravovať, iba deti vedie z pozície tzv. sprievodcu. Pracovné tempo si každý prispôsobí podľa vlastných schopností. Vyše 40 rokov experimentov umožnilo sformulovať 12 princípov, na ktorých je metóda založená [3]:

1. Budovanie schém

Schémou chápeme súbor známych informácií o nejakom objekte, mieste či emócii, ktoré prirodzene každý človek vytvára konkrétnymi udalosťami. Tento princíp hovorí o prepojení už známych poznatkov, ktoré umožňujú žiakom rozmýšľať o nových poznatkoch v už známom kontexte. Príkladom takejto schémy v Hejného metóde je napríklad autobus alebo rodinné vzťahy, ktoré sú pre deti známe a prirodzené, takže nové poznatky nadobudnuté v tejto schéme majú trvalú hodnotu.

2. Práca v prostredí

Hejného metóda úlohy vsádza do jednotlivých prostredí, ktoré sú žiakom známe z bežného života. Spomedzi 25 prostredí, ktoré metóda používa sú to napríklad: autobus, rodina, schody, ciferník alebo pavučiny. Veľkou výhodou pri práci v prostredí je najmä jasná formulácia jednotlivých úloh, ktoré sú väčšinou žiakom hneď jasné a motivácia, keďže vďaka prostrediam žiaci častokrát nemajú dojem, že sa práve učia, ale naopak, že sa hrajú.

3. Prelínanie tém

Ak sú žiaci schopní dať si jednotlivé poznatky do súvisu, dokážu si novo zistenú informáciu lepšie zapamätať a čo lepšie, niekedy aj sami odvodiť. Je teda potrebné vyhnúť sa učeniu jednotlivých faktov osobitne, pretože žiak tak má väčší problém si neskôr na informácie spomenúť.

4. Rozvoj osobnosti

Pri klasickej výučbe na základných školách sa častokrát stáva, že žiakom je názor vyučujúceho predstavený ako jediný správny. V dôsledku toho často žiak nie je schopný svoj vlastný názor rozvinúť a ani k tomu nie je motivovaný. Preto sa v Hejného metóde kladie dôraz na učenie argumentácie a diskusie, teda dávať žiakom možnosť, aby sami rozvinuli svoje chápanie o problematike, zatiaľ čo vyučujúci hrá rolu len ako akýsi pozorovateľ a usmerňovač.

5. Skutočná motivácia

Zrejme jedným z hlavných dôvodov, prečo je matematika vo všeobecnosti medzi žiakmi menej obľúbený predmet je, že žiakom už v rannom veku, kedy sú s jej základmi zoznámené, nepríde zaujímavá. Hodiny sú skôr takou rutinou, kde sa najprv odprezentuje daná látka a žiak následne musí prepočítat niekoľko takmer totožných príkladov. Žiaka, ktorý ešte stále vyhľadáva vo veciach hru takáto činnosť neosloví. Preto je potrebné úlohy vytvárať tak, aby žiak bol zvedavý, aké je správne riešenie, a prečo je práve toto riešenie správne. Následne je žiak schopný po nadobudnutí viacerých poznatkov jednoduchšie preniknúť do vyššej matematiky plnej abstrakcie.

6. Reálne skúsenosti

V Hejného metóde je veľmi dôležitým cieľom to, aby žiak nadobudol nové skúsenosti a nadobudol ich vlastnou činnosťou. Neskôr je schopný z týchto skúseností robiť všeobecné úsudky, a teda je potrebné, aby žiak úlohu riešil sám, pretože jedine tak si osvojí danú problematiku natoľko, že bude schopný riešiť aj iné úlohy. Ak je žiak postup úloh naučený alebo je mu riešenie ukázané, stráca motiváciu problematiku hlbšie pochopiť a nie je schopný vyriešiť iné úlohy sám.

7. Radosť z matematiky

Je potrebné v žiakovi prebudiť vnútornú motiváciu a záujem o danú problematiku, pretože tak po vyriešení úloh bude mať vnútorný pocit radosti, ktorá bude poháňať jeho motiváciu zaujímať sa ešte viac.

8. Vlastný poznatok

Poznatok získaný vlastnou činnosťou je oveľa hodnotnejší ako poznatok prevzatý. Žiak postupne zbiera radu poznatkov, ktoré v triede konzultuje so spolužiakmi, a spoločne kriticky rozmyšľajú, pričom každý sám dobre rozumie teórii, ktorú má, pretože na ňu prišiel vlastnou činnosťou. A teda pre hlbšie pochopenie akejkoľvek problematiky je potrebné sa na jej objasnení podieľať.

9. Rola učiteľa

V klasickej výučbe matematiky je rola učiteľa dominantná. Jeho úlohou je predať nejaké poznatky a tieto poznatky sa potom žiak snaží aplikovať pri riešení úloh. Hejného metóda k role učiteľa pristupuje úplne inak. Jeho autorita v triede je nižšia než v klasickom ponímaní. Síce má vedomosti na to, aby poznatky poskytol, nerobí tak a necháva žiakov, aby ich sami prezentovali. Pôsobí skôr ako tichý dozor, organizuje priebeh hodiny, zadáva úlohy a zvyšuje ich náročnosť, avšak nikdy neprezrádza ich riešenie.

10. Práca s chybou

Chybovať je pre človeka v akejkoľvek činnosti prirodzené a obzvlášť tomu tak je v procese učenia sa. Ak sa chyba správne analyzuje, dokáže rozsah nadobudnutého poznatku ešte umocniť. Nie len že analyzovaním príčin chyby sa vyhne zopakovaniu danej chyby, ale taktiež dokáže prehĺbiť chápanie danej problematiky a zlepšiť

kritické myslenie žiaka. Taktiež z psychologického hľadiska, ak človek chybuje, v jeho pamäti sa taká skúsenosť udrží dlhšie ako keby v úlohe uspel, a preto poznatok nadobudnutý správnou analýzou chyby dokáže byť aj časom lepšie uchopiteľný.

11. Primerané výzvy

V každej triede sa stretnú žiaci s rôznou úrovňou porozumenia matematiky. Je potrebné, aby výučba bola zameraná na všetky úrovne, a teda aby žiak ktorý je nadmerne nadaný nemrhal svojim talentom pri riešení úloh, ktoré už dávno vie riešiť, ale naopak aby sa im venovala špeciálna pozornosť a ponúkaním nových výziev a zadávaním náročnejších úloh tak aj naďalej rozvíjať ich potenciál. Rovnako tak treba špeciálne pristupovať aj k žiakom, ktorí sa javia v triede ako slabší. Takých nabádať k vysloveniu vlastných názorov a myšlienok a vyjadriť uznanie.

12. Podpora spolupráce

Každý žiak má svoje preferencie, keď príde na otázku, či chce pracovať samostatne alebo v skupine, a je potrebné aby tieto preferencie boli rešpektované. V Hejného metóde je práca v skupine viac než vítaná a diskusia je pri riešení úloh, kedy učiteľ nie je autoritou a nepovie správne riešenie, kľúčom k odhaleniu pravdy. Oproti klasickej výuke, kedy je spolupráca častokrát chápaná ako podvádzanie a je trestaná, Hejného metóda naopak nabáda k šíreniu poznatkov v kolektíve a konzultovanie ich správností.

1.2.1 Vzorová hodina vedená Hejného metódou

V rámci získavania poznatkov o princípoch Hejného metódy mi bolo umožnené zúčastniť sa aj na hodine na 1.súkromnom gymnáziu, Bajkalská 10 v Bratislave, vedenej konzultantkou tejto bakalárskej práce, Dagmar Môťovskou.

Po príchode žiakov do triedy som bola hneď svedkom dodržiavania dvanásteho princípu, princípu podpory spolupráce. Deti si automaticky posadali okolo veľkého stolu v strede miestnosti, a bolo zrejmé, že im je takéto rozmiestnenie úplne prirodzené. Následne učiteľka predstavila tému, ktorej sa na danej hodine mali venovať, a bolo vidno, že deti sú témou nadšené a tešili sa na prácu. Podľa môjho názoru tomu dopomohol spôsob akým učiteľka tému predstavila, a síce nie veľmi odborne a oficiálne, čo by žiakov len odradilo, no naopak ich v podstate vyzvala k hre. Tu sa v praxi ukázalo využitie ďalšieho

princípu, princípu skutočnej motivácie a ako čiastočný následok toho aj princípu radosti z matematiky.

Úlohou detí bolo vystrihovať a zlepovať nastrihané štvorce z papiera, na ktoré mali nakresliť útvary a predmety, ako napríklad okno, koberec či dvere a vytvoriť tak izbičku. Deťom úloha prišla veľmi hravá a ja a mala som pocit, že ani deti, nemali dojem že sa v skutočnosti niečomu novému učia. Učili sa avšak priestorovej predstavivosti. Tu je možno vidieť ďalšiu ukážku z princíпов, a to reálne skúsenosti. Deti si totiž samy skúsili zložiť izbičku, a ak im náhodou niektorá stena nesesedela, ako napríklad, že dvere na nej boli otočené hore nohami, veľmi rýchlo prišli na to, čo je zle a chybu boli schopné samy opraviť.

Nasledujúca úloha bola o čosi náročnejšia, no aby učiteľka zmiernila dojem, aký by deti mohli mať, predstavila úlohu pomocou krátkej rozprávky. Rozprávka bola o kocke, ktorá bola tak neforemná, že jej žiaden krajčír nevedel ušit šaty. V podobe rozprávky učiteľka dosiahla dodržanie princípu o práci v prostredí, a síce, žiakom predstavila situáciu ako príbeh, čo im napomohlo k lepšej vizualizácii problému, aký mali riešiť. Úlohu detí teda bolo ušit kocke také šaty a v preklade vystrihnúť zo štvorčekového papiera súvislý útvar tak, aby sa s ním dala zakryť každá stena kocky - niečo, čo je známe pod pojmom sieť kocky. Žiaci bez toho, aby vôbec tušili, akú neľahkú úlohu práve riešia s radosťou skúšali vystrihovať rôzne útvary a na moje podivenie sa väčšine podarilo úlohu vyriešiť pomerne skoro a dokonca niektorí žiaci ušili šiat aj viac - našli viacero sietí. Dokázali teda vyriešiť pomerne náročnú úlohu bez akéhokoľvek trápenia, naopak s radosťou.

Po celý čas bola učiteľka v úlohe pozorovateľa. Žiakov motivovala k práci, chválila za úspešné riešenie, zadávala nové, náročnejšie úlohy a v prvom rade, nepomáhala žiakom úlohu riešiť, čím zohľadnila princíp roly učiteľa, ktorý podľa Hejného metódy nie je natoľko dominantná osoba v triede ako počas klasickej výuky.

1.3 Využitie Hejného metódy v tvorbe edukačného softvéru

Pri tvorbe edukačného softvéru by sme okrem zachovania princíпов z časti 1.1 tejto práce mali zvoliť aj nejaký spôsob alebo metódu, akou chceme pomocou softvéru vzdelávať. Nasledovanie princíпов Hejného metódy sa ukázalo byť skvelým nástrojom pri tvorbe edukačného softvéru, keďže samotné princípy sa do veľkej miery prelínajú s istými požiadavkami

na edukačný softvér, čo jednoznačne zjednodušuje proces tvorby.

V prípade edukačného softvéru určeného najmä nižším vekovým kategóriám hrá veľkú rolu vizuálne prevedenie. Ak si dáme pútavé užívateľské prostredie ako cieľ pri tvorbe softvéru, Hejného metóda nám ponúka skvelý nástroj v podobe tzv. prostredí, krátko opísaných v druhom princípe - Práca v prostredí časti 1.2.1 tejto práce. Hejného metóda má týchto prostredí celkovo 25, a každé jedno slúži na vzdelávanie istého matematického celku. Nie len, že to dopomáha k vizuálnej hodnote užívateľského rozhrania, ale každé prostredie obsahuje pedagógmi vymyslené a overené modelové úlohy, ktoré slúžia ako predloha pri tvorbe samotného obsahu edukačného softvéru.

Ako Príklad tejto súhry technológie a Hejného metódy si uvedieme niekoľko bakalárskych prác venujúcich sa danej téme, ktoré vznikli pod vedením školiteľa tejto bakalárskej práce na Fakulte matematiky, fyziky a informatiky Univerzity Komenského.

1.3.1 Prostredie Biland

Táto edukačná aplikácia [5] bola vytvorená v roku 2019 Júliou Gáblikovou. Jej úlohou je hravou formou deti naučiť dvojkovej sústave, ktorú používajú počítače. V prostredí sa pracuje s grošmi, ktoré sú reprezentované ako textové reťazce podľa ich hodnoty: $Ag=1$, $Bg=2$, $Cg=4$, $Dg=8$, $Eg=16$... (každý groš je dvojnásobkom predošlého). Každé prirodzené číslo je možné zapísať tak, že každý typ grošu je možné použiť najviac jedenkrát. V dvojkovej sústave (alebo "bilandskom zápise") potom číslo napíšeme tak, že na mieste "jednotiek" je počet Ag grošov, na mieste "desiatok" počet Bg grošov (dvojok), na mieste "stoviek" je počet C grošov (štvoriek) atď.

1 je v Bilande Ag alebo 1

2 je v Bilande Bg alebo 10

3 je $Bg + Ag$ alebo 11

4 je Cg alebo 100

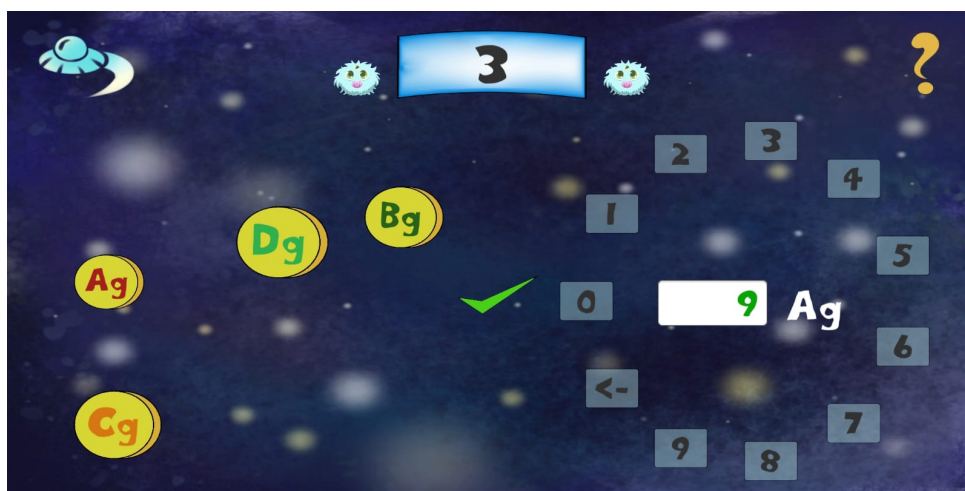
5 je $Cg + Ag$ alebo 101 ($1 \times 4 + 0 \times 2 + 1 \times 1$)

30 je $Eg + Dg + Cg + Bg$ alebo 11110 ($1 \times 16 + 1 \times 8 + 1 \times 4 + 1 \times 2 + 0 \times 1$) atď

[6]

Aplikácia sa skladá zo 4 úrovní série úloh, pričom v rámci jednej úrovne úlohy graduujú. V prvej úrovni si žiaci osvoja jednoduché základy premeny čísel z dvojkovej sústavy

v bilandskom zápise do zápisu v desiatkovej sústave. V druhej úrovni sa žiaci zoznámia so sčítaním a odčítaním čísiel v dvojkovej sústave v bilandskom zápise. V tretej úrovni sa žiaci naučia porozumieť prepisom bilandského zápisu čísla v dvojkovej sústave do tradičného zápisu čísla v dvojkovej sústave pomocou jednotiek a núl. A v poslednej štvrtej úrovni prichádza pre žiakov skutočná výzva a princípom úloh je znova práca so sčítaním a odčítaním čísiel v dvojkovej sústave, avšak nehladáme tentokrát výsledok, ten je daný. Niektoré cifry sú však vynechané a úlohou žiaka je nájsť všetky riešenie.



Obr. 1: Ukážka z aplikácie Bilandia

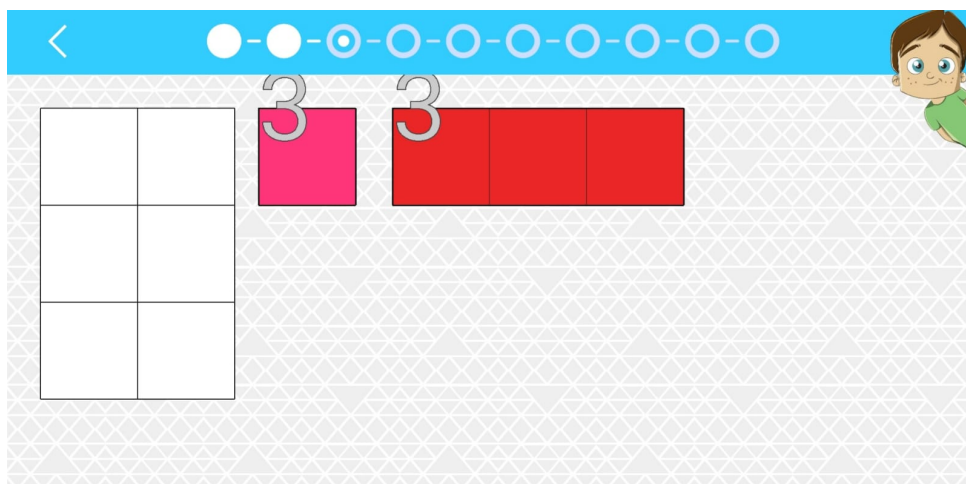
Na prvý pohľad mi ovládanie neprišlo úplne intuitívne, no po pár neúspešných klikoch som si ho osvojila a s aplikáciou sa dalo jednoducho pracovať. Grafické prevedenie je skutočne prepracované, autorka si dala záležať na postavičkách, ktoré pre deti určite pôsobia pútavo a splnila tým istotne požiadavku na edukačný softvér pre deti, ktorý by mal pôsobiť hravo. Taktiež sa autorke podarilo vyhnúť sa zbytočným textom pri nápovede či zadaní úlohy, ktoré nahradila nápovedou vo forme príkladu správne vyriešenej úlohy. Gradácia úloh mi prišla primeraná a taktiež sa mi páčila možnosť, ktorú ponúka štvrtá úroveň, a síce zoznámenia sa s úlohami, ktoré majú viacero riešení, čo je celkom nový koncept pre žiakov základnej školy.

1.3.2 Prostredie Parkety

Táto edukačná aplikácia vznikla v rámci bakalárskej práce v roku 2018 [7] a jej autorkou je Andrea Spišáková. Jej úlohou je pomôcť lepšie porozumieť rovinným útvarom, a to

pomocou skladania kusov parkiet na štvorcovú sieť, pričom pravidlom je, že sa parkety nesmú prekryvať.

Táto aplikácia má 4 úrovne gradujúcej série úloh, pričom cieľom v každej úlohe je poukladať parkety na sieť tak, aby bola celá pokrytá parketami a tie sa navzájom ne-prekrývali.



Obr. 2: Ukážka z aplikácie Parkety

Aplikácia sa mi osobne veľmi páčila, užívateľské prostredie je príjemné a veľmi intuitívne. Graficky pútavých prvkov, ktoré by podľa mňa deti ocenili je v porovnaní s aplikáciou Bilandia menej, no podľa môjho názoru to neuberá na celkovom dojme. Autorka veľmi dobre poňala myšlienku minimalizovať text a do aplikácie umiestnil panáčika, ktorý po kliknutí ponúka informácie relevantné k danej obrazovke, a síce návod ako riešiť danú úlohu či spôsob ako ovládať jednotlivé prvky. V rámci úrovni je zvolená veľmi prirodzená gradácia vo veľkosti štvorcovej siete, tvare kusov parkiet, vo vyšších úrovniach taktiež v počte jednotlivých kusov parkiet a taktiež sa tu stretávame s konceptom viacerých riešení danej úlohy.

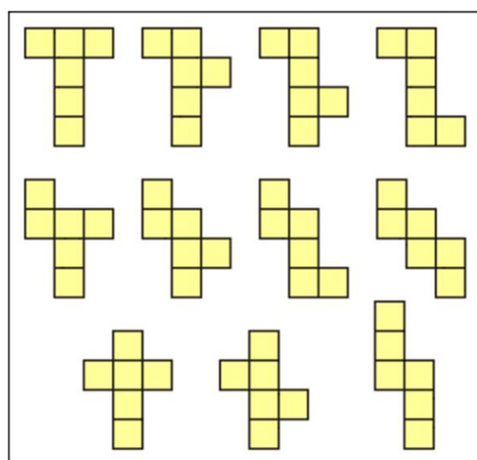
1.4 Prostredie Siete kocky

V tejto bakalárskej práci pri tvorbe softvéru využijeme znalosť prostredia Siete kocky. Prostredie Siete kocky je adresované žiakom od 2. ročníka a jeho hlavnou úlohou je rozvinúť predstavivosť v priestore a pochopiť prechod medzi 2D a 3D geometriou. Ako aj názov tohto prostredia napovedá, jeho úlohy sú primárne založené na pochopení pojmu

sieť kocky a práci s ňou, takže pre účely tejto práce sme sa rozhodli všetky úlohy založiť na podobnom princípe, a síce, úlohou používateľa je nájsť sieť zadanej kocky. Jednotlivé typy úloh si však popíšeme v nasledujúcej kapitole, avšak predtým je potrebné zdefinovať si pojem sieť kocky tak, ako ho budeme chápať v našom kontexte, aby sme tým čitateľa zbavili nejasností pri čítaní ďalších častí.

1.4.1 Sieť kocky

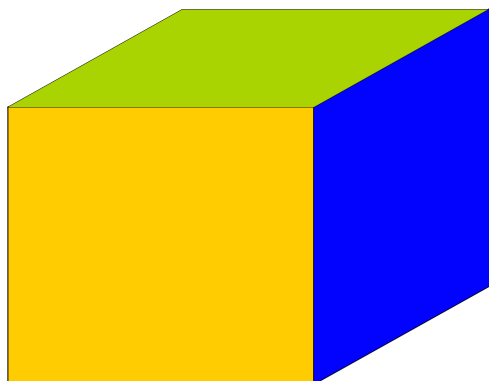
Podľa [8] sieťou n -stena rozumieme množinu mnohouholníkov zhodných s jeho stenami, vhodne umiestnených v rovine tak, že ich zložením dostaneme hranicu telesa. Táto definícia je dostatočná pre jednoduché teleso ako je kocka len v prípade, že neuvažujeme o tom, že by jednotlivé steny kocky neboli totožné. Počet takýchto sietí je všeobecne známy a postupu jeho zistenia ani dôkazu správnosti sa v tomto texte nebudeme venovať. Všetkých 11 sietí je ilustrovaných na obrázku 3, pričom sieť ktorá by vznikla rotáciou okolo stredu alebo sieť stredovo súmerná sa považuje za rovnakú.



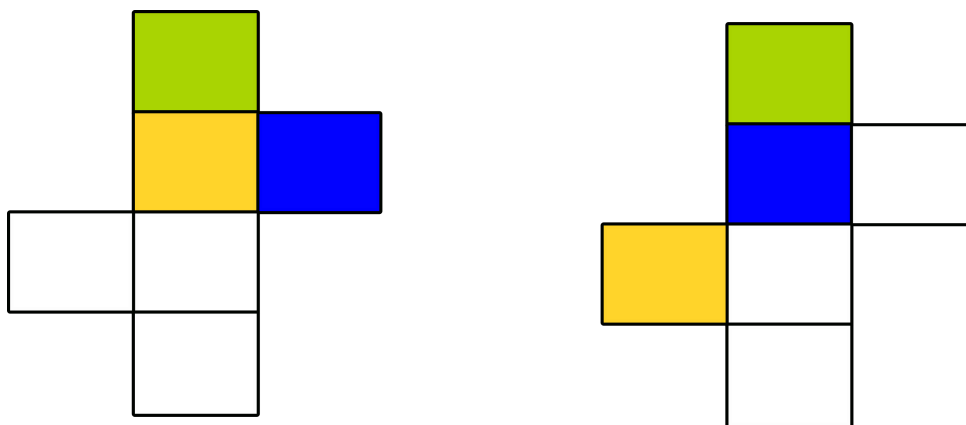
Obr. 3: Siete kocky

Pre potreby našej práce avšak potrebujeme sieť definovať pre trochu náročnejšiu formu kocky, a síce takú, že jej steny nebudeme uvažovať ako totožné, naopak každá stena bude jedinečná. To má za následok fakt, že kocku v našej práci máme možnosť istým spôsobom označovať - zafarbovať jej steny, resp. vrcholy, čím pevne určíme pozíciu steny v závislosti od ostatných. Uvedené je teda dôležité zobrať do úvahy pri vyčíslívaní všetkých možných sietí. Predstavme si kocku ako na obrázku 4 a pre zjednodušenie uvažujme, že všetky

zvyšné steny majú bielu farbu. Uvažujme tiež siete na obrázku 5.



Obr. 4: Kocka s označenými stenami



Obr. 5: Siete kocky s označenými stenami

Siete majú totožný tvar, avšak jednotlivé jej časti tú zafarbené rôzne. I keď by sa zdalo, že sa jedná o dve úplne odlišné siete, z oboch by sme boli schopný zložiť tú istú kocku. Uvedené slúži ako ilustrácia problému pri vyčísľovaní všetkých možností a implementácii jeho riešenia sa budeme venovať v kapitole 3 tejto práce.

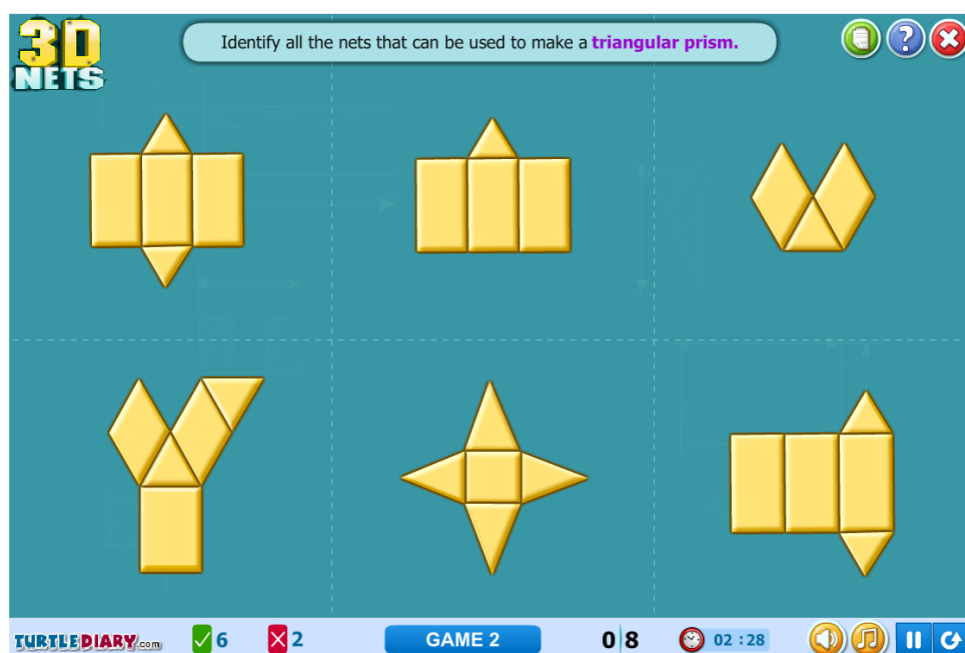
1.5 Podobné riešenia

Edukačný softvér už je súčasťou výučby dlhé roky, počas ktorých vzniklo nespočetne veľa aplikácií, ktoré sú určené pre mnohé účely - zatiaľ čo niektoré obsah látky vykládajú interaktívnym spôsobom, iné látku precvičujú. Ani výučba 2D a 3D geometrie nie je výnimkou, a tak už existuje niekoľko aplikácií, ktoré uľahčujú učenie sa tejto oblasti ma-

tematiky, no podľa môjho názoru je stále veľký priestor na zlepšenie. Spomenuté softvéry síce nie sú založené na Hejného metóde, no aspoň do istej miery pokrývajú látku, ktorou sa zaoberáme aj v prostredí Siete kociek.

1.5.1 Nets of 3D Shapes - Geometry Game

Hra je určená pre žiakov na precvičenie znalosti základných 3D objektov a ich sietí. Ponúka dva typy úlohy, a síce hádanie správneho názvu 3D útvaru a hľadanie 3D útvaru podľa názvu. Hra je v skutočnosti len zopakovaním si základných znalostí a neponúka žiadnu ďalšiu funkcionálnosť. Za pozitívum avšak považujem peknú vizualizáciu skladania siete útvarov do samotných útvarov, čo jednoznačne napomáha vytvoreniu lepšej predstavy o prechode z 2D geometrie do 3D geometrie.



Obr. 6: Ukážka z hry Nets of 3D Shapes - Geometry Game

1.5.2 Hrací kostka

Hra je súčasťou edukačného portálu matika.in, ktorého obsah je zameraný na vzdelávanie všetkých ročníkov základnej školy vo všetkých matematických oblastiach, s ktorými sa počas štúdia stretnú. Každá oblasť pozostáva z nejakej série úloh, a pre niektoré oblasti sú na portále taktiež vytvorené hry, ktoré precvičujú znalosti o niečo zábavnejšou formou.



Obr. 7: Ukážka z hry Hrací kostka

Ponuka pre výučbu 3D geometrie na stránke nie je veľmi podchytená a nachádza sa tam len jedna séria úloh s názvom Hrací kostka. V úlohách je potrebné správne doplniť sieť hracej kocky číslami na jej stenách tak, aby vznikla správna hracia kocka, a síce súčet na protiľahlých stenách bol rovný 7. Za pozitívum považujem poskytnutie akejkoľvek schémy v podobe hracej kocky, s ktorou sa hádam každý žiak niekedy stretol, no mám dojem, že daná látka má oveľa väčší potenciál a zaslúži si na portále viacero typov úloh.

1.6 Technológie

Než sa v nasledujúcich kapitolách začneme zaoberať návrhom riešenia a implementáciou, je potrebné spomenúť technológie, ktoré sme sa rozhodli pri vývoji použiť. Výsledkom mojej práce bude mobilná aplikácia pre zariadenia s operačným systémom Android, a preto sa prirodzene naskytla možnosť vývoju v Android Studio. S Android Studio mám už nejaké skúsenosti a po zadaní práce som si bola istá, že sa vydám touto cestou, avšak viacero výhod ma presvedčilo si nakoniec vybrať na vývoj Unity Engine.

Unity Engine

Unity Engine je game engine, ktorý slúži na vývoj hier a aplikácií rôznych zameraní pre rôzne platformy. Možnosť generovania riešenia pre viaceré platformy a na prvý pohľad náročný editor scén, ktorý však ponúka veľa možností úprav grafického rozhrania boli dôvody, prečo som sa preň rozhodla.

C#

Na to, aby sme dokázali jednotlivé objekty v aplikácii či v hre ovládať sú potrebné skripty. Unity Engine podporuje tvorbu skriptov v jazykoch JavaScript a C#. Keďže mám s ním predošlé skúsenosti a viac sympatizujem s objektovo orientovanými jazykmi, rozhodla som sa pre C#.

JetBrains Rider

Na úpravu skriptov v jazyku C# existuje niekoľko vývojových prostredí. Spomedzi ponúkaným sa však mne najviac zapáčil Rider od spoločnosti JetBrains. Nie len preto, že mám skúsenosti s viacerými produktami od tejto spoločnosti, a teda mi je ovládanie editora pomerne blízke, ale aj preto, že Rider ponúka veľkú podporu skriptovania práve pre Unity. Ako najväčší prínos považujem možnosť jednoduchého debugovania skriptov počas toho ako sú vykonávané v Unity editore, keďže Rider disponuje integrovanou obojsmernou komunikáciou. Za ďalšiu veľkú výhodu považujem prvotriednu analýzu kódu, ktorá upozorňuje na dodržiavanie konvencií písania C# skriptov. Rider taktiež ponúka jednu veľmi užitočnú funkcionality, a to analýzu výkonu a upozornenie na neefektívne používanie Unity API.

2 Návrh

Pred samotnou implementáciou riešenia je potrebné si čo najlepšie navrhnuť, ako má riešenie vyzerieť. Napomáha to ku organizovanosti práce a značne to zjednodušuje implementáciu. Ako je spomenuté v časti Teoretické východiská tejto práce, pri tvorbe edukačného softvéru je potrebné zohľadňovať niektoré zásady. Od začiatku práce na aplikácii je potrebné riešenie konzultovať s kompetentnými osobami, ktoré majú skúsenosti s cieľovou skupinou a dokážu tak poskytnúť objektívnu kritiku. V našom prípade bol návrh konzultovaný s RNDr. Dagmar Môťovskou, PhD., konzultantkou tejto práce, ktorá prispievala skúsenosťami zo školského prostredia a vedomosťami o Hejného metóde. Návrhu riešenia sa budeme venovať v tejto časti bakalárskej práce. Opíšeme ako sme aplikáciu rozčlenili do jednotlivých úrovní, vizuálne predstavíme riešenie grafického rozhrania a taktiež slovne opíšeme funkcionality jednotlivých častí aplikácie.

2.1 Logika gradácie úrovní

Na to, aby sme používateľa čo najlepšie oboznámili s problematikou, musíme začať s nižšou náročnosťou a postupne pridávať. Gradáciu medzi jednotlivými úrovňami sme teda založili na nasledujúcej úvahe.

Najprv používateľa zoznámime s kockou, ktorá má zafarbené steny a dáme mu takmer voľnú ruku vo výbere siete, ktorú pre vyriešenie použije. Avšak, ak by sme nechali používateľa zvoliť v každej úlohe rovnakú sieť, veľmi rýchlo by sme ho tým dovedli k núde a rovnako tak by sme nedosiahli cieľ našej aplikácie. Rozhodli sme sa tomu zabrániť tak, že používateľovi povolíme použitie rovnakej siete maximálne trikrát a pri opätovnej snahe použiť rovnakú sieť ho upozorníme vizuálne v scéne danej úlohy.

Neskôr v druhej úrovni úlohy používateľovi trochu sťažíme tým, že mu nedáme možnosť výberu akejkoľvek siete, no náhodne mu niektorú zadáme a jeho úlohou je s danou sieťou úlohu vyriešiť. Niekomu by sa mohlo zdať, že ho tým zbavíme polovice práce, no faktom je, že niektoré siete nie sú veľmi intuitívne, a tak sa náročnejšie predstavuje ich priemet do 3D geometrie. Sme teda toho názoru, že ak takto používateľa dovedieme k práci s viacerými sieťami, pomôžeme mu sa lepšie ponoriť do problematiky, ktorou sa práca zaoberá.

V ďalšej, tretej úrovni gradáciu zabezpečíme pomocou trocha iného spôsobu zafarbenia kocky. Namiesto stien v tretej úrovni totiž zafarbujeme jednotlivé vrcholy kocky. Takéto zafarbenie donúti používateľa pri vytváraní a farbení siete rozmýšľať aj nad tým, ktoré vrcholy siete kocky sa pri priemete spoja do jedného vrcholu kocky. Po vzore prvej úrovne nedovolíme používať rovnakú sieť viac ako trikrát, a to z dôvodu už spomenutého vyššie.

V poslednej štvrti úrovni analogicky s druhou znova používateľa chceme nabádať ku tomu, aby si osvojil premýšľanie na viacerých sieťach, a tak mu pri začiatku každej úlohy nejakú náhodne vyberieme.

Gradácia medzi úrovňami ide veľmi prirodzeným tempom a v princípe má používateľ za úlohu v každej úrovni to isté - nájsť sieť alebo pracovať so zadanou a zafarbiť ju tak, aby zodpovedala danej kocke, pričom úlohy komplikujeme hlavne zadanou kockou. Používateľa teda zbytočne neoznamujeme s iným typom úlohy a sme toho názoru, že takto si lepšie osvojí problematiku, ktorá nie je vôbec jednoduchá.

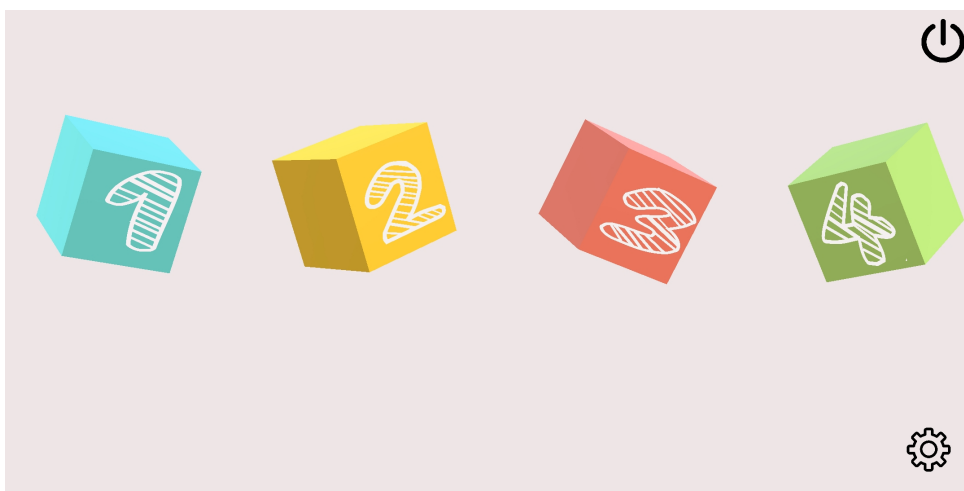
2.2 Grafické užívateľské rozhranie

Od začiatku práce sme prikladali vysoký dôraz na vizuálne prevedenie výsledného produktu. Jednalo sa o aplikáciu určenú pre deti, a tento fakt sme sa snažili zohľadniť. V prvom rade sme mali za cieľ, aby bolo ovládanie aplikácie jednoznačné, čo sme zabezpečili jednotnými ovládacími prvkami na jednotlivých obrazovkách - scénach. V druhom rade sme sa snažili jednotlivé scény zbaviť nepotrebných prvkov, ktoré by mohli pôsobiť rušivo a vsadili radšej na jednoduchšie, no jasné prevedenie. Následne sme jednotlivým prvkom určili jednotný dizajn, ktorý sme navrhli v súlade s odporúčaniami tvorby edukačného softvéru pre deti, a síce, že sme volili hravé a pútavé prevedenie. V nasledujúcich častiach sa budeme venovať jednotlivým scénam aplikácie a opíšeme si ich funkcionality s názornou vizuálnou ukážkou.

2.2.1 Hlavné menu

Pri tvorbe hlavného menu sme mali za cieľ užívateľa upútať základnými prvkami idey aplikácie, a tým bola kocka. Preto sme použili práve tento objekt ako označenie jednotlivých úrovní aplikácie. Efekt sme doplnili a animáciu nasledujúcu kliknutie na jednotlivú úroveň na obrazovke, a síce, rotáciu kocky okolo x-ovej osi po dobu dvoch sekúnd, ktorej

nasleduje prepnutie obrazovky na scénu danej úrovne. V našej aplikácii povolujeme otvoriť danú úroveň len v prípade, že je sprístupnená, čo zabezpečí zvládnutie ôsmich úloh z predchádzajúcej úrovne. Vizuálne znázorňujeme to, či je úroveň sprístupnená farebným prevedením. Sprístupnená úroveň je sfarbená na modro, žltu, červeno alebo zeleno, podľa úrovne v tomto poradí a nesprístupnená ja sfarbená na šedo. Ukážka scény na obrázku 8.



Obr. 8: Ukážka zo scény hlavného menu

Scéna hlavného menu je ďalej doplnená o tlačidlo s ikonou vypnutia, ktoré slúži na vypnutie aplikácie. Toto tlačidlo sme doplnili pre účely príjemnejšieho ovládania, keďže pred jeho implementáciou bolo možné aplikáciu vypnúť len cez manažéra úloh na zariadení, čo nám neprišlo vhodné pre našich cieľových užívateľov.

Posledným prvkom scény je tlačidlo nastavení. Po kliknutí na toto tlačidlo sa otvorí dialógové okno, v ktorom má užívateľ možnosť resetovať jednotlivé úrovne aplikácie. Ukážka dialógového okna na obrázku 9. Dialógové okno pozostáva zo štyroch ikoniek jednotlivých úrovní, ktoré obsahujú informáciu, koľko úloh danej úrovne má užívateľ na danom zariadení vyriešených. Po jednotlivými úrovňami sa nachádza tlačidlo ktoré slúži na reset danej úlohy. Pre zachovanie logiky o postupnosti riešenia a náročnosti daných úrovní sme zaručili, že reset konkrétnej úrovne zabezpečí aj reset všetkých vyšších úrovní.



Obr. 9: Ukážka zo scény hlavného menu

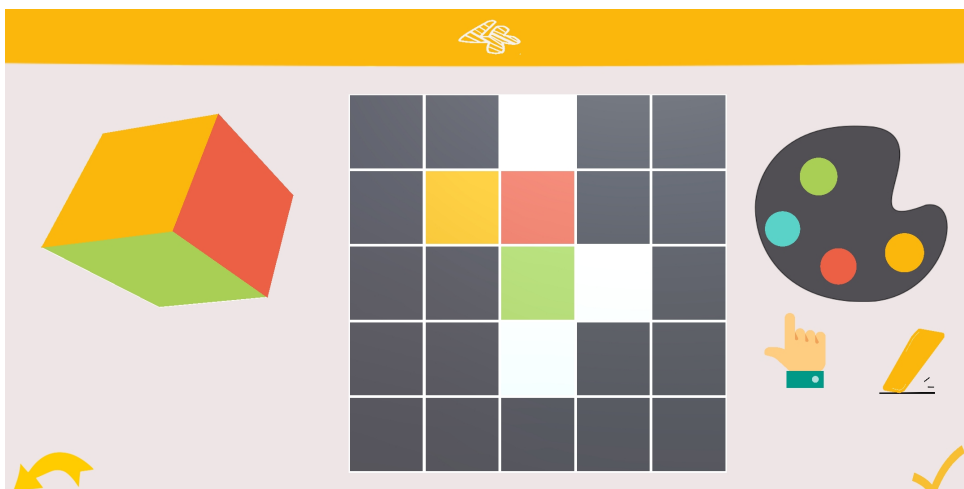
2.2.2 Prvá úroveň

Po otvorení prvej úrovne sa na obrazovke zobrazí scéna pozostávajúca z niekoľkých objektov, ktoré sú konštantné pre všetky úrovne. Medzi nimi je ukazovateľ čísla úlohy vrámci úrovne na vrchnej časti obrazovky. V ľavom dolnom rohu sa nachádza tlačidlo „späť“, kliknutím na ktoré je možný návrat do hlavného menu. V pravom dolnom rohu sa nachádza tlačidlo „fajka“, kliknutím na ktoré je možné overiť či je daná úloha dobre vyriešená.

V strede scény sa nachádzajú hlavné objekty pre danú úroveň. Vľavo sa nachádza kocka, ktorá sa dá otáčať okolo osi x aj y. Steny kocky sú zafarbené modrou, červenou, žltou, zelenou farbou alebo nie sú zafarbené vôbec.

V strede sa nachádza mriežka, ktorá reaguje na dotyk podľa toho, aký mód je práve zvolený. Po prvom otvorení úrovne je zvolený mód vytvárania siete kocky na mriežke. Opätovným kliknutím na označené políčko v mriežke sa políčko odznačí. Tento mód je opätovne možné dosiahnuť kliknutím na tlačidlo so symbolom ruky na pravej strane. Nad tlačidlom so symbolom ruky sa nachádza ikona palety so štyrmi farbami - červenou, modrou, zelenou a žltou. Kliknutím na jednotlivé farby sa mód prepne z označovania políčok na mriežke na zafarbovanie políčok v mriežke príslušnou farbou, pričom je možné zafarbiť len políčko, ktoré bolo predtým označené. Vedľa tlačidla sa nachádza aj tlačidlo so symbolom gumičky, po kliknutí na ktoré sa prepne mód na gumovanie zafarbených políčok, a teda v tomto móde je možné políčko zo zafarbeného zmeniť na označené. Ukážka scény

na obrázku 10.

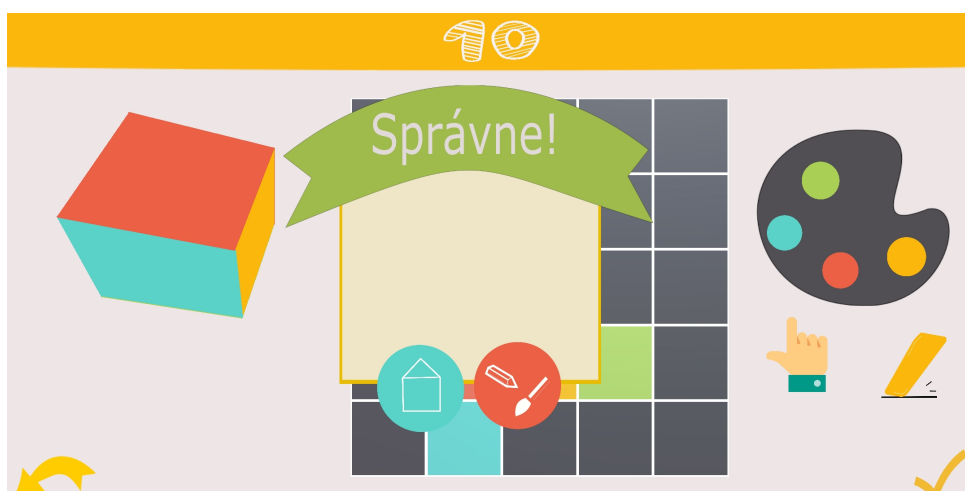


Obr. 10: Ukážka zo scény prvej úrovne

V princípe je úlohou užívateľa v mriežke označiť políčka tak, aby tvorili sieť danej kocky. Užívateľ je limitovaný, koľkokrát môže označiť políčka v rovnakom zoskupení, a teda koľkokrát môže v mriežke označiť rovnakú sieť. V prípade, že sa snaží použiť rovnakú sieť nad povolený limit, scéna na to upozorní zafarbením pozadia mriežky žltou farbou po dobu 2 sekúnd a negatívnym zvukovým signálom.

V prípade, ak kliknutím na tlačidlo „fajka“ užívateľ skontroluje svoje riešenie, ktoré nie je správne, scéna na to upozorní zafarbením pozadia mriežky červenou farbou po dobu 2 sekúnd a negatívnym zvukovým signálom. V opačnom prípade sa zobrazí v strede scény dialógové okienko so správou „Správne!“, zaznie pozitívny zvukový signál a po dvoch sekundách sa na obrazovke načíta ďalšia úloha pre danú úroveň. V prípade, že ide o poslednú desiatu úlohu úrovne, na obrazovke sa zobrazí dialóg so správou „Správne!“. Pod správou sa nachádzajú dve tlačidlá - tlačidlo „domov“, kliknutím na ktoré sa užívateľ dostane na hlavné menu a tlačidlo „editor“, kliknutím na ktoré sa zobrazí na obrazovke scéna editora prvej úrovne, ktorú spomenieme v ďalšej časti. Ukážka dialógového okna na obrázku 11.

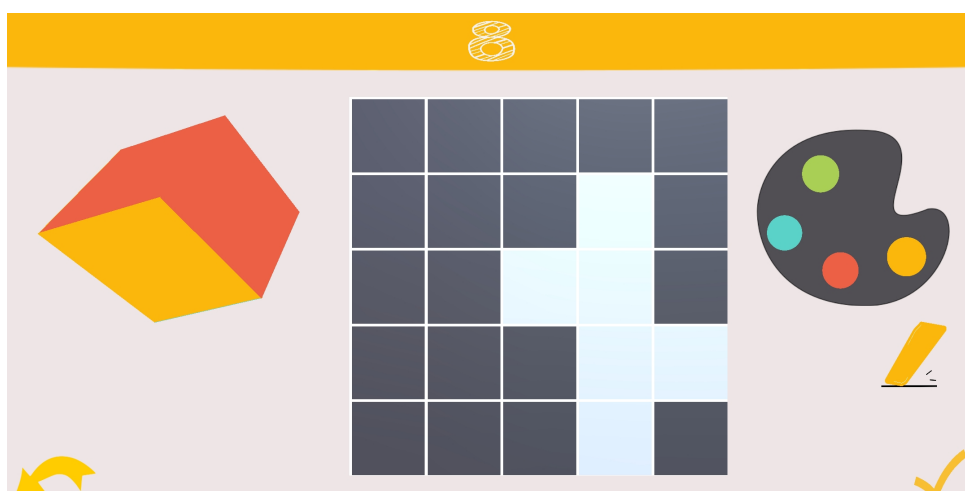
V prípade vlastnej úlohy prvej úrovne editovanej v editory scéna neobsahuje číselné označenie úlohy v hornej časti. Po úspešnom, resp. neúspešnom vyriešení úlohy sa scéna správa rovnako, ako keď užívateľ rieši poslednú desiatu úlohu úrovne.



Obr. 11: Ukážka zo scény prvej úrovne

2.2.3 Druhá úroveň

Druhá úroveň sa od prvej úrovne líši len v jednej veci, a síce, že pre túto úroveň nie je umožnený mód vytvárania siete kocky na mriežke, a teda sa na scéne nenachádza tlačidlo so symbolom ruky. Taktiež je pozmenená mriežka, a to tak, že už pri otvorení úrovne sú niektoré políčka označené. Tieto políčka teda nie je možné odznačovať, je možné ich len zafarbiť a farbu z nich zmazať rovnakým spôsobom ako v prvej úrovni. Ukážka scény na obrázku 12.

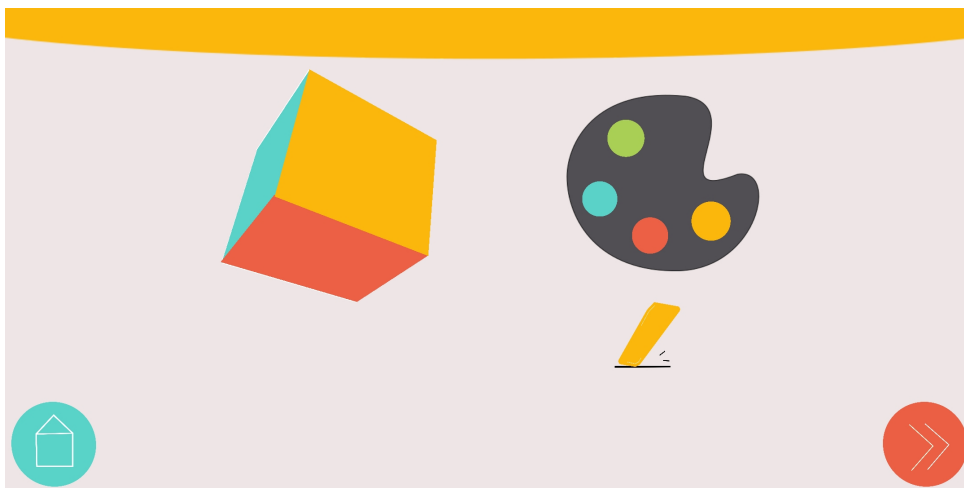


Obr. 12: Ukážka zo scény druhej úrovne

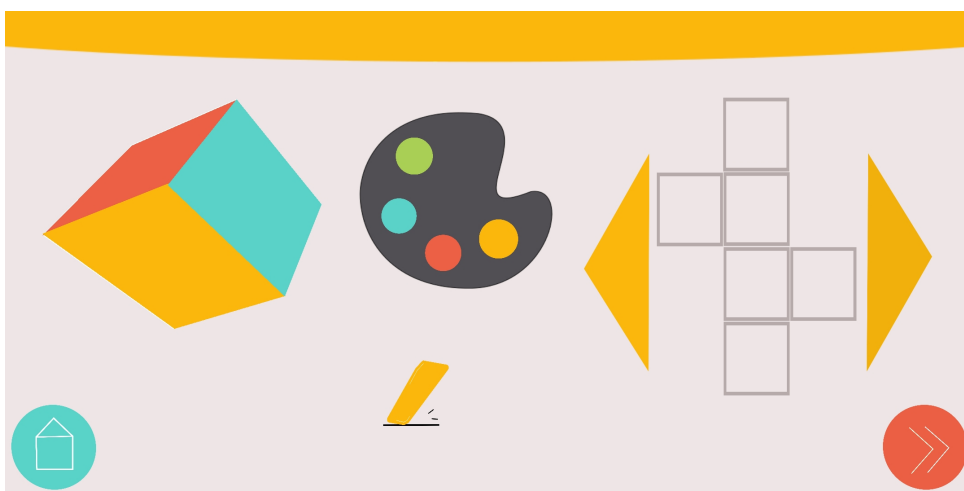
2.2.4 Editor prvej a druhej úrovne

V scéne editora prvej úlohy sa nachádzajú ovládacie prvky, ktoré majú totožné správanie s ovládacími prvkami v editore všetkých úrovní. V ľavom dolnom rohu sa nachádza tlačidlo "domov", ktoré umožňuje návrat na hlavné menu a v pravom dolnom rohu sa nachádza tlačidlo „ďalej“, ktoré umožňuje zahrať si práve editovanú vlastnú úlohu. V strede scény sa nachádza nezafarbená biela kocka, ktorou je možné otáčať rovnako ako v scéne prvej úrovne. Vedľa kocky sa nachádzajú dva prvky rovnaké ako v scéne úlohy - paleta s modrou, červenou, žltou a zelenou farbou a tlačidlo so symbolom gummy. Kocka reaguje na kliknutie a podľa toho, či je scéna v móde farbenia po kliknutí na niektorú z farieb na palete alebo v móde gumovania po kliknutí na tlačidlo s ikonou gummy. V móde farbenia sa po kliknutí na kocku príslušná stena zafarbí danou farbou a v móde gumovania sa stena odfarbí. Ukážka editora na obrázku 13.

V prípade editora druhej úrovne sa na scéne okrem všetkých prvkov scény editora prvej úrovne nachádza ešte možnosť výberu siete, s ktorou má byť daná úloha riešená. Ukážka editora na obrázku 14.



Obr. 13: Ukážka zo scény editora prvej úrovne

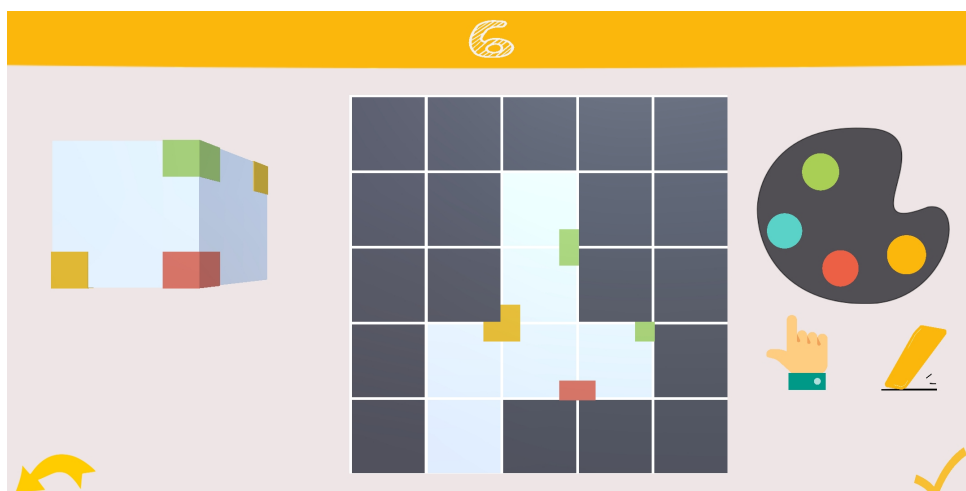


Obr. 14: Ukážka zo scény editora druhej úrovne

2.2.5 Tretia úroveň

Scéna tretej úrovne na prvý pohľad vyzerá obdobne scéne prvej úrovne. Jediné, v čom sa líši je kocka, ktorá v prípade tretej úrovne nemá zafarbené steny, avšak jej rohy sú zafarbené jednou z farieb - modrá, červená, zelená a žltá alebo nie sú zafarbené vôbec.

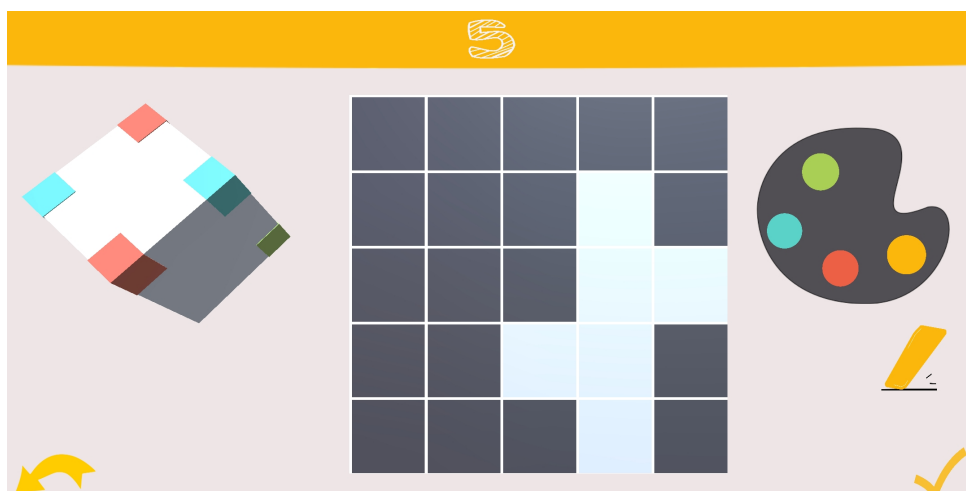
Podľa takejto kocky je upravená aj funkcionálna zafarbovania políčok mriežky po kliknutí na niektorú z farieb na ikone palety, je totiž možné zafarbiť rohy políčok na scéne, pričom po kliknutí na daný roh políčka sa roh zafarbí danou farbou a rovnako tak aj rohy príslušné rohy príľahlých označených políčok. Obdobné je správanie v gumovacom móde, a síce, po kliknutí na roh políčka sa daný roh a príslušné rohy príľahlých políčok odfarbia. Zvyšné prvky a funkcionálna je zhodná s prvou úrovňou. Ukážka scény na obrázku 15.



Obr. 15: Ukážka zo scény tretej úrovne

2.2.6 Štvrtá úroveň

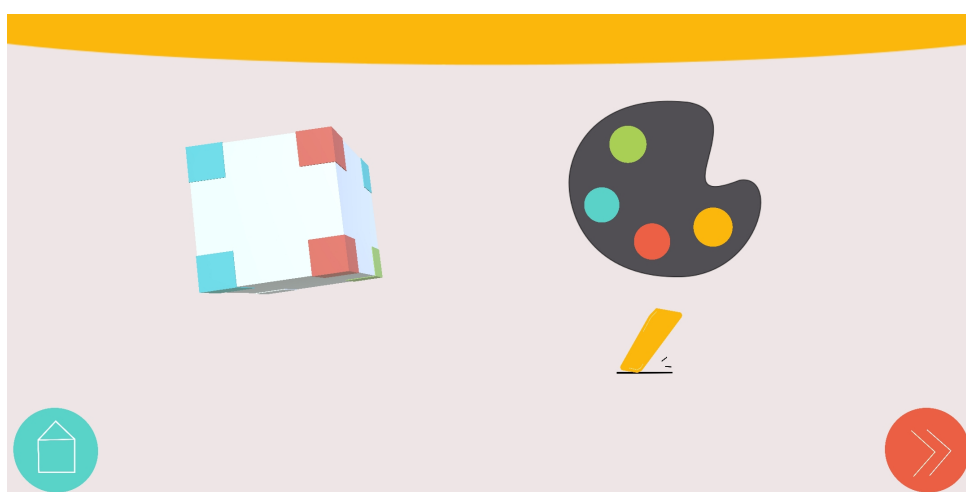
Scénu štvrtej úrovne môžeme opísať ako kombináciu scény tretej úrovne a scény druhej úrovne. V podstate kopíruje funkčnosť tretiu úroveň, avšak s jedným rozdielom analogickým ku druhej úrovni. Rovnako ako v druhej úrovni nie je totiž umožnený mód vytvárania siete kocky a na scéne sa nenachádza tlačidlo so symbolom ruky a mriežka už pri otvorení úlohy štvrtej úrovne sú políčka označené v zoskupení reprezentujúcom sieť kocky. Tieto políčka teda nie je možné odznačovať, je možné len zafarbiť ich rohy a farbu z nich zmazať rovnakým spôsobom ako v tretej úrovni. Ukážka scény na obrázku 16.



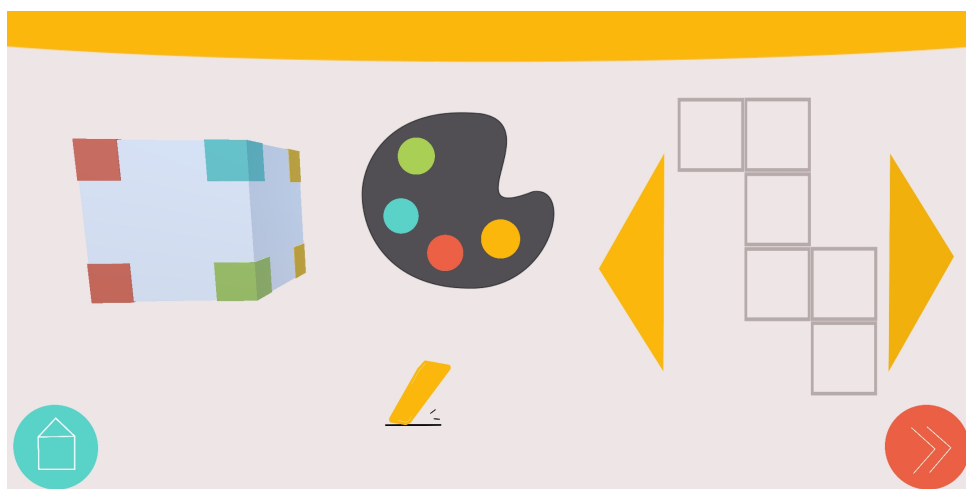
Obr. 16: Ukážka zo scény štvrtej úrovne

2.2.7 Editor tretej a štvrtej úrovne

Scéna editora tretej a štvrtej úrovne obsahuje takmer všetky prvky totožné s editorom prvej a druhej úrovne, len s jedným rozdielom. Užívateľ v tomto prípade nemá možnosť editovať steny, ale vrcholy danej kocky, a to obdobným spôsobom ako v editore prvej a druhej úrovne, výberom príslušnej farby a kliknutím na daný vrchol. Gumovanie farby z vrcholu je možné v móde gumovania, po kliknutí na tlačidlo s ikonou gummy. Ukážky scén na obrázkoch 17 a 18.



Obr. 17: Ukážka zo scény editora tretej úrovne



Obr. 18: Ukážka zo scény editora štvrtej úrovne

3 Implementácia

Po návrhu v procese vývoja aplikácie prirodzene dochádza ku fáze implementácie. V tejto časti vývoju sme museli venovať pozornosť niekoľkým veciam, a síce, chceli sme, aby naša aplikácia dokázala do istej miery nahradiť školácke učebnice a stále dokázala pomôcť danú látku pochopiť. Avšak ak by sme v aplikácii ponúkli len súbor niekoľkých príkladov, a po ich vyriešení by žiak mal prácu hotovú asi by taká aplikácia nemala žiadnu pridanú hodnotu oproti tej klasickej učebnici. Preto teda naša aplikácia umožňuje generovanie náhodných úloh, čím zabezpečíme, že žiak má možnosť precvičovať si znalosti na stále nových príkladoch. S tým prichádza na rad aj nutnosť zamyslieť sa nad tým, ako chceme vyhodnocovať riešenie týchto náhodných úloh, keďže uloženie výsledku každej možnej úlohy sa javí ako nerozumná a hlavne náročná úloha. Potrebujeme teda nástroj, ktorým by sme vedeli o akomkoľvek riešení, pre akúkoľvek úlohu rozhodnúť, či je správna. V nasledujúcich častiach tejto kapitoly sa bližšie pozrieme, ako spomenuté problémy implementujeme.

3.1 Logika generovanie úloh

- **Prvá úroveň**

Pri generovaní úloh prvej úrovne sa opierame o ideu, že čím viac stien kocky je zafarbených a čím viac farieb je pri tom použitých, je úloha náročnejšia. Preto sme sa rozhodli postupne pridávať počet stien a farieb a to v nasledujúcich krokoch:

- V prvej úlohe zafarbíme 2 rôzne steny rovnakou farbou
- v druhej úlohe zafarbíme 3 steny a tentokrát pri tom použijeme 2 rôzne farby.
- V tretej až piatej úlohe zafarbíme 4 steny a to s použitím 3, 4 a 4 rôznych farieb v tomto poradí pre každú úlohu.
- V šiestej a siedmej úlohe zvýšime počet zafarbených stien o jednu a použijeme najprv 3 farby v šiestej a následne 4 farby v siedmej úlohe.
- Nakoniec v siedmej až poslednej desiatej úlohe zafarbíme všetkých 6 stien a použijeme na to prv 3 farby a následne dvakrát 4 farby v posledných dvoch úlohách úrovne.

- **Druhá úroveň**

Generovanie úloh v druhej úrovni je založené na generovaní náhodnej kocky a v generovaní náhodnej siete. Generovanie kocky je v druhej úrovni totožné s prvou úrovňou, a teda v prípade druhej úrovne sa pridáva len náhodné vybratie jednej z 20 sietí kocky (symetrické siete rátame ako rozdielne), pričom zabezpečíme, že sa v rámci jednej hry v úrovni nemôže žiadna sieť zopakovať.

- **Tretia úroveň**

Tretia úloha pracuje s obdobnou ideou ako prvá úloha, a síce v tomto prípade, čím viac vrcholov zafarbíme a čím viac farieb pri tom použijeme, tým viac používateľovi úlohu sťažíme. Analogicky teda s pravidlami pri generovaní kocky v prvej úrovni sa generuje kocka aj v tretej úrovni, s tým rozdielom, že steny nahradíme vrcholmi.

- **Štvrtá úroveň**

V poslednej úrovni úlohy generujeme analogicky s druhou úrovňou s jediným rozdielom, a to tým, že kocku generujeme podľa pravidiel tretej úrovne.

3.2 Vyhodnocovanie úloh

V úvode tejto kapitoly spomíname, že je nutné vyriešiť kontrolu nájdených riešení jednotlivých úloh dynamicky, a teda nezávisle od danej úlohy. V tejto časti sa teda bližšie pozrieme na to, akým spôsobom sme sa rozhodli uvedené implementovať. V prvom rade si však lepšie definujme problém, ktorým sa budeme zaoberať, aby sme čitateľovi pomohli lepšie porozumieť riešeniu.

Našou úlohou je rozhodnúť, či sieť nájdená používateľom a vyznačená v mriežke v scéne danej úlohy zodpovedá nejakej sieti zadanej kocky. V prípade prvej, resp. druhej úrovne je riešenie tohto problému o niečo jednoduchšie ako v prípade tretej, resp. štvrtej úrovne. Práve preto sme sa v tomto prípade rozhodli problém rozdeliť na dva podproblémy, čím dokážeme v mnohých prípadoch skôr vyhodnocovanie ukončiť pre nesprávne riešenia.

Prvým problémom je zistenie, či sa z útvaru v mriežke dá poskladať akákoľvek kocka. Zatiaľ neporovnávame sieť s našou kockou, len zisťujeme, či má potenciál sa ňou stať. V prípade tretej a štvrtej úrovne totiž častou chybou pri riešení je, že používateľ zle

porozumie, ako sa pri skladaní kocky jednotlivé vrcholy siete spoja do vrcholu kocky. Dochádza tak k tomu, že nimi nájdené riešenie by v skutočnosti kockou byť nemohlo. Tým, že v prvom kroku vyhodnocovania najprv overíme túto skutočnosť vieme skôr používateľa upozorniť, že jeho riešenie nie je správne. Samozrejme, že z pohľadu používateľa je táto zmena nebadateľná, avšak z implementačného hľadiska má veľký význam, ak zbytočne nevykonávame zvyšok algoritmu.

Druhým problémom je overenie, či sieť skutočne zodpovedá zadanej kocke. Zatiaľ čo v prvej, resp. druhej úrovni budeme porovnávať jednoducho farby jednotlivých častí siete, v tretej, resp. štvrtej úrovni budeme musieť skontrolovať každý vrchol siete.

Jednoduchšie riešenie prvého aj druhého problému nám umožní vhodná reprezentácia objektov, ktoré do vyhodnocovania vstupujú, a síce kocky a mriežky s označenou sieťou, ktorú si objasníme v ďalšej časti.

3.2.1 Reprezentácia objektov

V prípade kocky sme sa dohodli na jednotnom značení a očíslovali sme si jednotlivé steny číslami od 0 po 5 v tomto poradí: predná stena, pravá stena, zadná stena, ľavá stena, dolná stena a horná stena. Samozrejme, že v aplikácii sa kocka dá otáčať, a tak nemá pevne určené, ktorá stena je otočená ktorým smerom, avšak v programe ostáva jej reprezentácia nemenná. Výsledkom teda bolo, že kocku reprezentujeme ako jednorozmerné pole šiestich prvkov. Prvky poľa sa však líšia v prípade prvej, resp. druhej úrovne a tretej, resp. štvrtej úrovne. V prvom prípade si v poli uchováваме jednoducho farby, ktorými sú steny zafarbené, zatiaľ čo v druhom prípade uchováваме reťazec vytvorený z farieb vrcholov prislúchajúcich danej stene v určenom poradí. Predstavme si dve kocky, pričom prvá z nich má prednú stenu označenú zelenou farbou, zadnú modrou farbou a zvyšné steny má biele. Zatiaľ čo druhá kocka má pri pohľade na prednú stenu jej pravý horný vrchol označený zelenou farbou a ostatné bielou. Potom ich reprezentácia je nasledovná : prvá kocka je reprezentovaná ako {zelená, biela, modrá, biela, biela, biela} a druhá kocka ako {„BZBB“, „ZBBB“, „BBBB“, „BBBB“, „BBBB“, „BBZB“}, kde B je biely vrchol prislúchajúci stene a Z je zelený vrchol prislúchajúci stene.

V prípade mriežky je reprezentácia celkom prirodzená, a síce dvojrozmerné pole prvkov - políčok mriežky. Analogicky s reprezentáciou kocky si pre prvú a druhú úroveň ukladáme

farby jednotlivých políčok a v prípade tretej a štvrtej úrovne reťazec vytvorený z farieb vrcholov prislúchajúcich jednotlivým políčkam. Jednotná reprezentácia kocky a mriežky nám neskôr v algoritme uľahčuje vyhodnocovanie.

Ako sme v úvode tejto práce popísali, počet sietí kocky je konečný a pomerne malý. Práve preto sme sa rozhodli pre zjednodušenie algoritmu uchovávať jednotlivé siete v programe na pevno. Napríklad:

```
1     private readonly string[] [] _net1Mapping = {
2         new [] { "", "ABCD", "" },
3         new [] { "ADEF", "DCGE", "CBHG" },
4         new [] { "", "EGHF", "" },
5         new [] { "", "FHBA", "" } };
6
7     private readonly string[] [] _net2Mapping = {
8         new [] { "HDAF", "DCBA", "CGEB" },
9         new [] { "", "ABEF", "" },
10        new [] { "", "FEGH", "" },
11        new [] { "", "HGCD", "" } };
12    ...
```

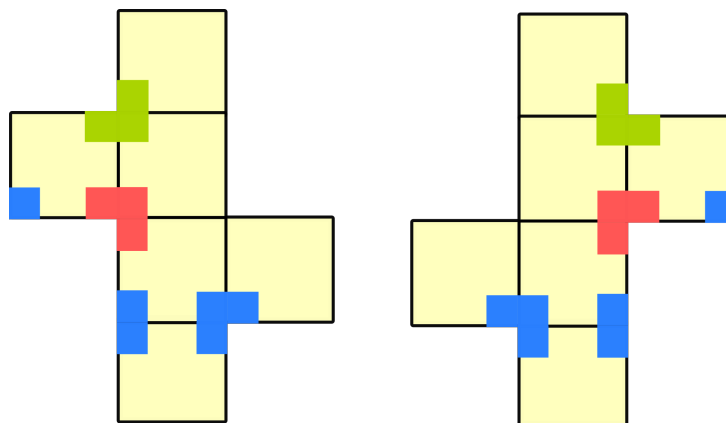
Takúto reprezentáciu jednotlivých sietí sme vybrali zámerne, keďže v prípade tretej a štvrtej úrovne potrebujeme dodatočnú informáciu o jednotlivých prvkoch. Potrebujeme totiž vedieť, ktoré vrcholy siete sa spoja do jedného vrcholu kocky. Preto sme sa rozhodli anotovať každý vrchol písmenom abecedy a následne sme schopní sieť reprezentovať ako dvojrozmerné pole, kde je každý prvok reťazcom vytvoreným z písmen abecedy podľa toho, ktoré vrcholy políčku prislúchajú. Prázdne reťazce v poli predstavujú políčka, ktoré sú prázdne.

3.2.2 Algoritmus

Samotný algoritmus sme rozdelili do niekoľkých krokov.

V prvom rade potrebujeme na mriežke nájsť sieť, ktorú náš používateľ označil ako riešenie. V úvode práce sme uviedli vzťah medzi sieťami, ktoré sú navzájom rotáciou okolo stredu alebo zrkadlovým obrazom, a teda je potrebné akceptovať aj takéto riešenie. Pri zrkadlových horizontálnych alebo vertikálnych otočeniach dochádza k netriviálnemu

preusporiadaniu označenia vrcholov, ktoré nie je možné jednoznačne určiť pre všetky siete. Ak by sme si predstavili kocku, ktorej sieťou je na obrázku 19 vľavo, zistili by sme, že zrkadlový obraz tejto siete nezodpovedá tej istej kocke. Muselo by dôjsť k zmene umiestnenia označených vrcholov.



Obr. 19: Symetrické siete

Rozhodli sme sa teda symetrické siete uchovávať v programe v rovnakej reprezentácii ako všetkých 11 sietí uvedenej vyššie a nezaoberať sa nimi na úrovni mriežky. Rotácie okolo stredu avšak zachovávajú pozície vrcholov, a teda budeme rátať len so 4 možnými rotáciami mriežky okolo stredu.

Pre každú mriežku budeme postupovať rovnako, a síce, že v mriežke nájdeme niektorú zo sietí v reprezentácii spomenutej vyššie. V prípade prvej, resp. druhej úrovne porovnanie prebieha len na základe porovnania prítomnosti nejakého prvku (neprázdny reťazec korešponduje s neprázdny políčkem v mriežke a prázdny s neoznačeným). V prípade tretej, resp. štvrtej úrovne využijeme tento krok zároveň pre kontrolu správnosti označenia, čím vyriešime prvý problém spomenutý v úvode tejto podkapitoly. Vrcholy anotované rovnakým písmenom teda musia byť označené rovnakou farbou, resp. neoznačené. Nie lenže v tomto kroku nájdeme umiestnenie siete v mriežke, zistíme aj ktorú z jedenástich sietí si používateľ zvolil. To nám umožní zjednodušiť nasledujúci krok.

Teraz keď už máme nájdenu sieť, zistíme, či je skutočne sieťou našej kocky. Správnosť porovnania sme sa rozhodli zabezpečiť nájdením všetkých správnych možností, v ktorých sa pokúsime nájsť používateľovo riešenie. V úvode práce sme sa zoznámili s problémom vyčíslenia všetkých možností siete kocky, s ktorou v našich úlohách pracujeme. Tento

problém sme sa rozhodli vyriešiť tak, že si v prvom rade v programe uložíme inštrukcie na vytváranie jednotlivých sietí, pričom využijeme označenie jednotlivých stien kocky spomenuté vyššie. Napríklad:

```
1      private readonly int [][] _net1Instructions = {
2          new [] { -1, 5, -1 },
3          new [] { 3, 0, 1 },
4          new [] { -1, 4, -1 },
5          new [] { -1, 2, -1 } };
6
7      private readonly int [][] _net2Instructions = {
8          new [] { 3, 0, 1 },
9          new [] { -1, 4, -1 },
10         new [] { -1, 2, -1 },
11         new [] { -1, 5, -1 } };
12         ...
```

Každé nezáporné číslo zodpovedá stene kocky, a teda vieme určiť, ktorá stena kocky sa stane akým políčkom siete. Takto avšak dokážeme nájsť práve jednu sieť, no my potrebujeme aj všetky možné zafarbenia (ilustrované na obrázku 5). To jednoducho docielime tak, že sieť na základe týchto inštrukcií vytvoríme pre každé otočenie kocky. Budeme teda simulovať otáčanie kocky do všetkých 24 možných natočení a v každom natočení vytvoríme siete podľa inštrukcií. Keďže sme v predošlom kroku zistili, ktorú sieť používateľ zvolil, pre každé otočenie nám stačí použiť len inštrukcie pre tú danú sieť. Výsledný počet možností bude maximálnym počtom možností a v niektorých prípadoch sa jednotlivé možnosti môžu zhodovať. To avšak platí len pre špeciálne prípady ako napríklad kocka, ktorá má prednú, pravú, zadnú a ľavú stenu označené zelenou farbou a hornú a dolnú stenu modrou farbou. Takéto prípady avšak vďaka spôsobu označovania stien pri generovaní úloh v našej aplikácii nemáme.

Teraz, keď už máme všetky správne možnosti a používateľovo riešenie, sme schopný jednoduchým porovnávaním dvojrozmerných polí zistiť, či je riešenie správne.

Na záver je treba spomenúť, že prvý krok tohto algoritmu vykonávame len v prípade prvej, resp. tretej úrovne, keďže práve v tých má užívateľ možnosť výberu siete. V ostatných prípadoch je sieť vybraná náhodne aplikáciou a teda si program pamätá, kde a

akú sieť používateľovi zadal.

Zoznam použitej literatúry

- [1] Karolčík, Š.: Návrh koncepcie hodnotenia kvality edukačne zameraného digitálneho obsahu a softvérov. Dostupné na webovskej stránke: http://files.virtual-lab.sk/APVV%20VV-11/2012/Koncepcia_hodnotenia_e_obsah_edusoft_navrh_karolcik_v2.pdf.
- [2] Krnáč, J. -Sudolská, M. -Trajtel', L., Učiteľova dielňa. Dostupné na webovskej stránke: http://www.statpedu.sk/files/sk/o-organizacii/projekty/projekt-dvui/publikacie/ucitelova_dielna.pdf
- [3] Co je to „Hejného metoda“? Dostupné na webovskej stránke: <https://www.h-mat.cz/hejneho-metoda>
- [4] 12 kľúčových princípů. Dostupné na webovskej stránke [:https://www.h-mat.cz/principy](https://www.h-mat.cz/principy)
- [5] Gáblíková, J.: Softvérová podpora vyučovania matematiky Hejného metódou - prostredie Biland, FMFI UK Bratislava, 2019
- [6] Málková, P., Příručka pro rodiče žáků s výukou matematiky podle metody prof. Milana Hejného. Dostupné na webovskej stránke: <https://ucebnice.fraus.cz/file/edee/2015/05/prirucka-pro-rodice3.pdf>
- [7] Spišáková, A.: Softvérová podpora vyučovania matematiky Hejného metódou - prostredie Parkety, FMFI UK Bratislava, 2018
- [8] Medek, V. a kol.: Matematická terminológia, SPN Bratislava, 1975