**Andrea** **Spišáková**

**2018**

**SOFTVÉROVÁ PODPORA VYUČOVANIA MATEMATIKY HEJNÉHO METÓDOU – PROSTREDIE PARKETOVANIE**

Bakalárska práca

**UNIVERZITA KOMENSKÉHO V BRATISLAVE  
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY**

Študijný program: Aplikovaná informatika  
Študijný odbor: 2511 Aplikovaná informatika  
Školiace pracovisko: Katedra aplikovanej informatiky  
Školiteľ: RNDr. Peter Borovanský, PhD.  
Konzultant: RNDr. Dagmar Môťovská, PhD.

**Andrea** **Spišáková**

**2018**

**SOFTVÉROVÁ PODPORA VYUČOVANIA MATEMATIKY HEJNÉHO METÓDOU – PROSTREDIE PARKETOVANIE**

Bakalárska práca

**UNIVERZITA KOMENSKÉHO V BRATISLAVE  
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY**

75643039



Univerzita Komenského v Bratislave

Fakulta matematiky, fyziky a informatiky

**ZADANIE ZÁVEREČNEJ PRÁCE**

|  |  |
| --- | --- |
| **Meno a priezvisko študenta:** Andrea Spišáková  **Študijný program:** aplikovaná informatika (Jednoodborové štúdium, bakalársky  I. st., denná forma)  **Študijný odbor:** aplikovaná informatika  **Typ záverečnej práce:** bakalárska  **Jazyk záverečnej práce:** slovenský  **Sekundárny jazyk:** anglický | |
| **Názov:** | Softvérová podpora vyučovania matematiky Hejného metódou - prostredie Parketovanie  *The Hejny method educational software - environment "Parqueting"* |
| **Anotácia:** | Aplikácie sa opierajú o didaktickú kvalitu vyučovania matematiky Hejného metódou, zadania úloh budú vyberané z učebníc matematiky Hejný, M., Jirotková, D., Slezáková, J., Bomerová, 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 gradujúcich 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. |
| **Vedúci:** | RNDr. Peter Borovanský, PhD. |
| **Konzultant:** | RNDr. Dagmar Môťovská, PhD. |
| **Katedra:** | FMFI.KAI - Katedra aplikovanej informatiky |

**Vedúci katedry:** prof. Ing. Igor Farkaš, Dr.

|  |  |  |
| --- | --- | --- |
| **Dátum schválenia:** | 16.10.2017 | doc. RNDr. Damas Gruska, PhD.  garant študijného programu |
| študent |  | vedúci práce |

75643039



Univerzita Komenského v Bratislave

Fakulta matematiky, fyziky a informatiky

**Dátum zadania:**

10.10.2017

Čestné prehlásenie

Podakovanie

Abstrakt

Abstract

# Obsah

[Úvod 12](#_Toc503449059)

[1 Teoretické východiská 13](#_Toc503449060)

[1.1 Edukačný softvér 13](#_Toc503449061)

[1.2 Hejného metóda vyučovania matematiky 14](#_Toc503449062)

[1.3 Prostredie Parketovanie 16](#_Toc503449063)

[1.4 Podobné existujúce riešenia 17](#_Toc503449064)

[1.4.1 Hardvérové riešenia 18](#_Toc503449065)

[1.4.2 Úlohy z matematiky pre deti na základných školách 19](#_Toc503449066)

[1.4.3 Matemág - Dobrodružná edukačná hra 21](#_Toc503449067)

[1.5 Predchádzajúce bakalárske práce 22](#_Toc503449068)

[1.5.1 Násobilkové štvorce 22](#_Toc503449069)

[1.5.2 Hadíky 23](#_Toc503449070)

[1.5.3 Autobus 24](#_Toc503449071)

[1.6 Technológie 25](#_Toc503449072)

[1.6.1 Android 25](#_Toc503449073)

[1.6.2 Android Studio 25](#_Toc503449074)

[1.6.3 Java 26](#_Toc503449075)

[Zdroje 27](#_Toc503449076)

# Zoznam obrázkov

[Obrázok 1: Povolené tvary parkiet v našej mobilnej aplikácii 16](#_Toc503448628)

[Obrázok 2: Príklad úlohy v prostredí Parketovanie 17](#_Toc503448629)

[Obrázok 3: Príklad úlohy v prostredí Parketovanie 17](#_Toc503448630)

[Obrázok 4: Papierové parkety s magnetmi a magnetickou doskou 18](#_Toc503448631)

[Obrázok 5: Drevené parkety 19](#_Toc503448632)

[Obrázok 6: Úlohy z matematiky pre deti na základných školách - riešenie úlohy 20](#_Toc503448633)

[Obrázok 7: Úlohy z matematiky pre deti na základných školách - prehľad nášho riešenia 21](#_Toc503448634)

[Obrázok 9: Matemág - jednoduchá úloha na ceste za čarodejníkom 22](#_Toc503448635)

[Obrázok 10: Násobilkové štvorce 23](#_Toc503448636)

[Obrázok 11: Hadíky 24](#_Toc503448637)

[Obrázok 12: Autobus 25](#_Toc503448638)

# Úvod

Žijeme v dobe obrovského pokroku a rozkvetu informačných technológií. Sú všade okolo nás – v masmédiách, zdravotníctve, doprave či administratíve. Školstvo nie je žiadnou výnimkou a vyučovanie pomocou tabletov, počítačov či interaktívnych tabúľ je dnes bežné, ba priam nevyhnutné. Aj najmladší žiaci ovládajú prácu s nimi bez väčších problémov, a práve to nám umožňuje tvoriť mobilné aplikácie, ktoré rozširujú obsah pracovných zošitov a spestrujú priebeh vyučovacej hodiny.

Našou aplikáciou sa pokúsime prispieť do rozrastajúcej sa softvérovej podpory pre vyučovanie matematiky Hejného metódou. Zameriavame sa na žiakov prvého stupňa základnej školy, ktorí sa vyznačujú zvedavosťou, hravosťou a chuťou objavovať. Naším cieľom je vhodne spracovať prostredie Parketovanie do digitálnej formy, ktorá bude lákavá nielen po vizuálnej stránke, ale najmä po stránke edukačnej. Našimi možnými konkurentmi sú drevené, plastové, či papierové verzie Parketovania. Chceme, aby naša edukačná aplikácia bola minimálne rovnako zaujímavá a dobre hrateľná. Preto budeme generovať stále nové a rozdielne úlohy, ktorých náročnosť sa bude stupňovať. Navyše, úlohy nebudú len jedného typu, ale vytvoríme štyri skupiny úloh s malými obmenami. Napokon umožníme žiakom vytvoriť si vlastnú úlohu a vyriešiť ju. Okrem toho zabezpečíme aj príjemné používateľské prostredie a ľahké, intuitívne ovládanie.

Veríme, že naša mobilná aplikácia bude prínosom jednak pre školy a učiteľov, ale aj pre samotných žiakov, ktorí budú môcť, podľa zásad Hejného matematiky, sami objavovať matematické zákonitosti hravou formou. Matematika tak pre nich nebude strašiakom, ale práve naopak – bude priestorom pre samostatné objavovanie a bádanie, ktoré je tým najlepším prostriedkom na získavanie vedomostí.

# Analýza problematiky

V tejto kapitole si ozrejmíme pojem *edukačný softvér* a analyzujeme teoretické východiská, ktoré sú základom pre našu aplikáciu a bez ktorých takýto softvér nemožno tvoriť. Priblížime si teda princípy Hejného metódy vyučovania matematiky a tiež samotné prostredie Parketovanie. Nemenej dôležité a prínosné sú už existujúce hardvérové a softvérové riešenia. Všimneme si ich výhody, ktorými sa môžeme inšpirovať, ale aj nevýhody a chyby, ktorým sa snažíme vyhnúť. Podobne sa prizrieme aj na predchádzajúce bakalárske práce, ktoré sa tiež zaoberali tvorbou softvéru založeného na princípoch Hejného metódy vyučovania matematiky. Napokon sa oboznámime s technológiami, ktoré budeme pri našej práci používať.

## Edukačný softvér

Edukačný softvér je softvér vytvorený špeciálne pre vyučovanie ako nástroj pre učiteľov na učenie alebo pre študentov na učenie sa [1]. Zaujímavým je aj ďalší pohľad na tento druh softvéru. Na kvalitný edukačný softvér sa pozeráme ako na múdry papier (na ploche obrazovky), ktorý nám pomáha experimentovať, manipulovať s objektmi, objavovať vzťahy a zákonitosti, skúmať a konštruovať [2].

Okrem toho, že edukačný softvér poskytuje zaujímavé prostredie nejakej hry, používateľ od neho prirodzene očakáva aj pohodlné ovládanie, príjemné používateľské prostredie a interakciu.

Z [1] vyberáme niektoré z mnohých otázok, ktoré si neustále treba klásť pri vyvíjaní takéhoto softvéru

1. z hľadiska výučby:

* Aký je účel nášho softvéru? Kde sa bude používať? Pre koho je určený?
* Aká je validita softvéru, t.j. učí softvér to, čo má učiť, čo učiteľ očakáva a potrebuje?
* Je korektný po odbornej stránke? Je jeho obsah v súlade s osnovami?
* Zodpovedá náročnosť zvolenému okruhu používateľov?
* Je určený na individuálne použitie alebo podporuje spoluprácu?
* Poskytuje spätnú väzbu?
* Je softvér interaktívny?

1. z hľadiska používateľa:
   * Je softvér jednoducho ovládateľný? Podobá sa iným aplikáciám, má štandardné fungovanie základných klávesov a tlačidiel? Dá sa jednoducho naučiť používať?
   * Má príťažlivý vzhľad?
   * Je kvalita zvuku, farieb a obrázkov dostatočná?
   * Je stabilný a odolný voči zlým vstupom?
   * Je lokalizovaný do slovenčiny, respekt. materinského jazyka?

## Hejného metóda vyučovania matematiky

Vít Hejný, pedagóg a matematik českého pôvodu, veril, že zákonitosti matematiky možno deti učiť aj inak. Alebo skôr nechať deti objavovať ich samé. Počas svojho pedagogického pôsobenia prerušovaného udalosťami 2. svetovej vojny sformuloval princípy, ktoré neskôr doplnil a upravil jeho syn Milan Hejný v spolupráci so svojimi kolegami. Túto prácu nebrali ako povinnosť, ale ako poslanie rozvíjať samostatnosť v myslení žiakov. Hoci sa ich snaha spočiatku nestretla s podporou vtedajšieho režimu, aj napriek mnohým prekážkam nezanikla a dnes sa teší obľube na mnohých školách. Učiteľ sa tu chápe ako rovnocenný partner, ktorý môže dopomôcť, naviesť, vysvetliť... K metóde tiež nepochybne patrí 12 základných princípov [3]:

* *Budovanie schém – Dieťa vie aj to, čo sme ho nenaučili*

Úlohy posilňujú schémy, ktoré máme v hlave. Prepájajú ich a pomáhajú vyvodzovať konkrétne úsudky.

* *Práca v prostrediach – Učíme sa opakovanou návštevou*

25 vypracovaných prostredí je deťom známych, a tak nie sú rozptyľované novými a neznámymi vecami. Prostredia sa líšia, pokrývajú rozličné oblasti matematiky, zahŕňajú rozličné štýly učenia a sú vypracované adekvátne k veku žiakov.

* *Prelínanie tém – Matematické zákonitosti neizolujeme*

Deti rôzne okruhy neberú ako samostatnú jednotku, ale vedia si navzájom prepojiť matematické javy a odhaliť ich súvislosti.

* *Rozvoj osobnosti – Podporujeme samostatné uvažovanie detí*

Metóda zachádza aj za hranice matematiky a snaží sa deti naučiť samostatnosti nielen pri riešení úloh v škole, ale aj v bežnom živote. Učí ich argumentovať, obhájiť svoje tvrdenie, ale aj niesť následky za svoje konanie.

* *Skutočná motivácia – Keď „neviem“ a „chcem vedieť“*

Úlohy sú postavené tak, aby deti do riešenia úloh nebolo treba nútiť. Sú motivované zvnútra a samé chcú objavovať a byť úspešné.

* *Reálne skúsenosti – Staviame na vlastných zážitkoch dieťaťa*

To, čo sme zažili na vlastnej koži, je nám bližšie a pomáha nám to ľahšie a rýchlejšie vyvodiť si všeobecný úsudok. Úlohy pripomínajú práve takéto každodenné zážitky a sú akousi analógiou bežného života.

* *Radosť z matematiky – Výrazne pomáha pri ďalšej výučbe*

Úspech pri riešení úloh motivuje a podnecuje naďalej pracovať zanietene a s radosťou.

* *Vlastný poznatok – Má väčšiu váhu než ten prevzatý*

Lepšie ako učiť sa poučky a definície je vyskúšať si to. Dieťa sa hlbšie zamýšľa nad tým, čo robí a ako to robí, než pri povrchovom učení sa o čomsi abstraktnom a neuchopiteľnom.

* *Rola učiteľa – Sprievodca a moderátor diskusií*

Učiteľ neprednáša výklad o matematike, učiteľ je pomocník a usmerňuje žiakov. Je rovnocenným partnerom, ktorí vedie žiakov správnym smerom, pomáha a podporuje ich rast.

* *Práca s chybou – Predchádzame zbytočnému strachu detí*

Učíme sa aj na vlastných chybách a spraviť chybu nie je hanba. Deti majú šancu nájsť prípadnú chybu v svojom riešení, opraviť ju a poučiť sa z nej.

* *Primerané výzvy – Pre každé dieťa zvlášť podľa jeho úrovne*

Slabší žiaci nie sú zbytočne odradení tým, že nedokážu vyriešiť žiadnu úlohu. A naopak, tí šikovnejší žiaci nemusia čakať a môžu riešiť aj zložitejšie a náročnejšie úlohy. Úloh je dosť pre každého. Žiakov neporovnávame vzhľadom k ostatným žiakom, ale vzhľadom k ním samým.

* *Podpora spolupráce – Poznatky sa rodia vďaka diskusii*

Keď deti zdieľajú svoje myšlienky, svojim spolužiakom nielenže pomáhajú s hľadaním riešenia, ale môžu im ho priblížiť aj zrozumiteľnejšie v jazyku svojich rovesníkov, než by to dokázal učiteľ.

## Prostredie Parketovanie

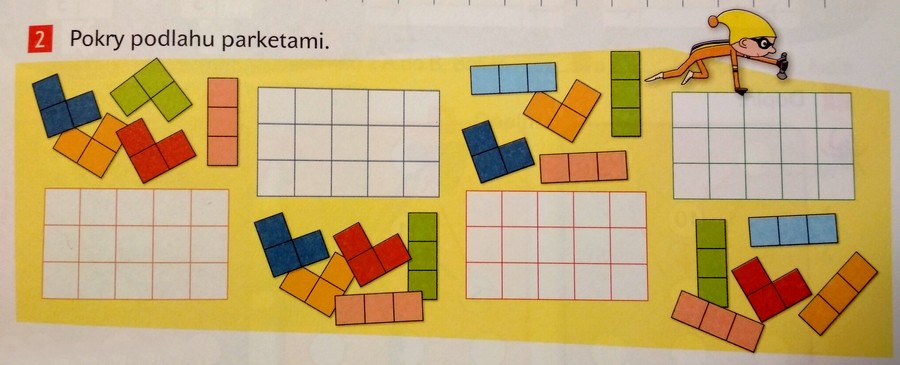
Hejného metóda pracuje s desiatkami prostredí, ktoré pokrývajú rôzne okruhy matematiky. Príkladom sú Súčtové trojuholníky, Krokovanie, Autobus, Cyklotrasa a iné. V našej mobilnej aplikácii sa venujeme len prostrediu Parketovanie.

V prostredí Parketovanie je daná štvorcová sieť predstavujúca podlahu a niekoľko druhov rozličných parkiet. Úlohou hráča je pokryť podlahu parketami, ktoré ma k dispozícii. Parkety sa dajú otáčať okolo svojho stredu. Parketu nie je možné preklopiť okolo niektorej osi. Hoci sú úlohy, ktoré povoľujú aj parkety iných tvarov, v našej aplikácii budeme mať práve 9 druhov parkiet. Tie sú znázornené na obrázku 1.

## povolene parkety

*Obrázok 1: Povolené tvary parkiet v našej aplikácii (zľava: Mono, Duo, Tri jednotky, Ľavé elko, Pravé elko, Zetko, Roh, Štvorka, Zobáčik)*

Úlohy v tomto prostredí sú zamerané na prácu s rovinnými útvarmi. Deti sa učia pracovať a orientovať v 2D priestore. Spoznávajú súmernosť a s ňou spojené otáčanie a symetriu. Rovnako je to práca s obsahom (počtom malých štvorčekov) a obvodom (okrajom plochy). Počet všetkých možností ako položiť parkety zase súvisí s kombinatorikou. Nemenej podstatné sú tvorivosť a predstavivosť, ktoré toto prostredie tiež rozvíja. Príklady úloh a ich rôznych obmien z učebnice [3] v prostredí Parketovanie sú uvedené na obrázku 2 a na obrázku 3.



*Obrázok 2: Príklad úlohy v prostredí Parketovanie*



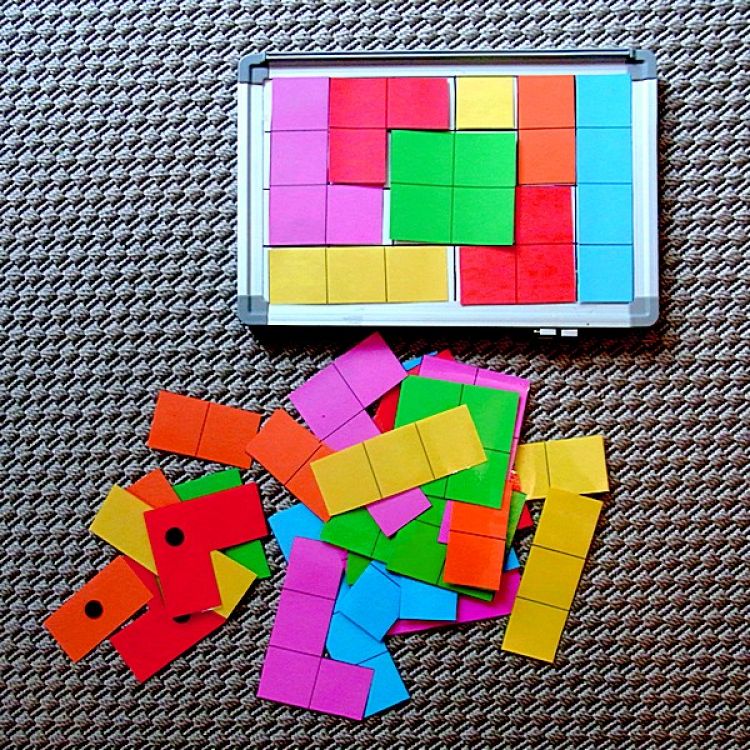
*Obrázok 3: Príklad úlohy v prostredí Parketovanie*

## Podobné existujúce riešenia

V tejto časti sa bližšie pozrieme na materiálne didaktické pomôcky, ale aj na už existujúce softvérové riešenia. Hoci v softvérovom podaní existujú stovky edukačných hier, zatiaľ len málo z nich je založených na Hejného metóde. My sa pozrieme práve na tie z nich, ktoré akceptujú pravidlá tejto metódy a pokrývajú úlohy z niektorého prostredia.

### Materiálne didaktické pomôcky

Vzhľadom k tomu, že prostredie Parketovanie obsahuje naozaj len dve podstatné položky – štvorcovú sieť a niekoľko parkiet, je naozaj ľahké vytvoriť si pomôcky svojpomocne, napríklad z papiera, alebo ich vylepšiť a pridať malú magnetickú dosku. Pre tých menej šikovných je možnosť zakúpiť si drevené alebo umelohmotné pomôcky. Na obrázku 4 a obrázku 5 vidíme ako takéto pomôcky môžu vyzerať [4][5].



*Obrázok 4: Papierové parkety s magnetmi a magnetickou doskou*



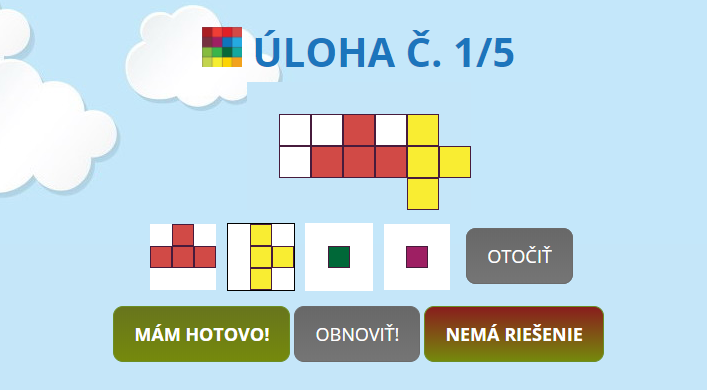
*Obrázok 5: Drevené parkety*

Práca s takýmito pomôckami je naozaj lákavá a zábavná. Ich výhodou je ľahká manipulácia a pekné farebné prevedenie. Je prirodzené, že deti sú viac zaujaté niečím, čo môžu chytiť do rúk, presúvať, otáčať. Za jedinú nevýhodu považujeme to, že pri nedbalej manipulácii nemusia parkety presne sadnúť do štvorcovej siete a okopírovať ju. Navyše tiež hrozí, že deti preklopia parkety okolo niektorej osi, čo nie je dovolené. Tieto zásady treba dodržiavať a žiakov na ne dôkladne upozorniť.

### Úlohy z matematiky pre deti na základných školách

Webová aplikácia [6] plná matematických úloh pre deti na základnej škole. Úlohy sú vypracované podľa Hejného metódy a pokrývajú mnohé prostredia, napr. Autobus, Hady, Susedia, Pavučiny. Sú rozdelené podľa náročnosti pre deti v jednotlivých ročníkoch. Úloh je naozaj neúrekom a sú rôznorodé.

Samotné prostredie Parketovanie však nie je spracované veľmi dobre. Výhodou sú pekné farby, príjemné prostredie pre deti. Ovládanie ale vôbec nie je intuitívne. Kurzor pri nadídení nad štvorcovú sieť zmení tvar a vytvára dojem, že túto sieť treba ťahať. Čo by sme skôr očakávali je ťahanie parkiet z ich pozície na sieť. To však nie je možné. Treba najskôr kliknúť na parketu, ktorú chceme ťahať a následne na miesto v štvorcovej sieti, na ktoré ju chceme vložiť. Ak chceme niektorú parketu otočiť, musíme na ňu najskôr kliknúť a potom kliknúť na tlačidlo Otočiť, ktoré si spočiatku nevšimneme. Aj keď parketu možno použiť iba raz, aj po jej vložení do siete stále zostáva v ponuke parkiet. Máme teda pocit, že ju môžeme použiť viackrát, no výberom však len premiestnime tú parketu, ktorá sa už nachádza na sieti. Parkety môžeme ukladať aj tak, že pretŕčajú cez okraje siete alebo sú mimo siete a dajú sa ukladať aj navzájom cez seba. Záleží na tom, či to povoľujeme, no zdá sa nám to neintuitívne a proti pravidlám. Navyše, funkčných tlačidiel je akosi priveľa, každé je inej farby, ktorá zdanlivo nijako nesúvisí s jeho významom. Dokonca tlačidlá pôsobia akýmsi mohutným dojmom a zdajú sa byť väčšie ako samotná štvorcová sieť a parkety. Iba tlačidlo na obnovenie hry a na otočenie sú rovnaké i keď nerobia to isté.



*Obrázok 6: Úlohy z matematiky pre deti na základných školách - riešenie úlohy*

Po prejdení všetkých úloh v danej sade sa dostávame k vyhodnoteniu. Ak sa nám podarilo správne vyriešiť aspoň jednu úlohu, sme síce pochválení za náš výkon, ale nemáme šancu opraviť si zlé riešenia, čo je jedným z princípov Hejného metódy. Hoci sada obsahovala len päť úloh, zobrazuje sa celkové hodnotenie so základom desať, čo menšie deti môže zmiasť. Taktiež sa objavia zrejme naše riešenia, no ako vidieť na obrázku 7, nezobrazujú sa správne a opäť sa mení kurzor, ak ho na ne položíme. Po kliknutí sa však nič neudeje.



*Obrázok 7: Úlohy z matematiky pre deti na základných školách - prehľad nášho riešenia*

### Matemág - Dobrodružná edukačná hra

Matemág je edukačná hra [7], ktorá využíva mnoho hovoreného slova. Hra má príbeh, v ktorom sa dvaja súrodenci pomocou matematiky snažia prekonať prekážky na ceste za čarodejníkom. Počas tejto cesty však musia čeliť úlohám z rôznych prostredí Hejného metódy vyučovania matematiky. Môžeme si pozrieť príbeh, ktorý vysvetľuje ako sa deti stretli s čarodejníkom, alebo sa rovno pustiť do plnenia úloh. Úlohy sú najskôr úplne jednoduché a ich náročnosť sa stupňuje.

Za pozitívum považujeme naozaj pekné farebné spracovanie, obrázky a hovorený text, ktorý nabáda hráča, čo robiť ďalej. Ovládanie je tiež intuitívne a hra dopomáha k jeho osvojeniu. Samotná hra je chytľavá a zaujme nielen deti, ale aj staršieho hráča. Súrodenci prechádzajú krajinou a cestou plnia rôzne matematické úlohy, ktoré ich posúvajú ďalej do cieľa. Ak si chceme opäť zahrať niektorú úroveň, matematické úlohy v nej sú vždy generované znova, a teda sú pre nás nové.



*Obrázok 8: Matemág - jednoduchá úloha na ceste za čarodejníkom*

## Predchádzajúce bakalárske práce

Existuje niekoľko predchádzajúcich bakalárskych prác, ktoré sa venovali edukačnému softvéru ako takému. My sa však pozrieme na tri vybrané bakalárske práce, ktoré sa rovnako ako naša, zakladajú na princípoch Hejného matematiky a transformujú niektoré zo známych prostredí do softvérovej podoby.

### Násobilkové štvorce

Bakalárska práca [8] vytvorená v roku 2017. Jej autorom je Martin Sadloň a pokrýva prostredie Násobilkové štvorce, teda násobenie a delenie čísel do 100. Úlohou je vhodne dopĺňať chýbajúce čísla do riadkov a stĺpcov.

Aplikácia je pekne spracovaná po grafickej stránke. Celkové ovládanie hry je tiež veľmi pohodlné. Každá obrazovka obsahuje len minimum potrebných prvkov, ktoré sú navyše jednoznačné a farebne oddelené. Po správnom vyriešení úlohy sa okolie hracej plochy rozsvieti na zeleno. V prípade nesprávneho riešenia je to červená farba. Hráč tak ihneď dostáva spätnú väzbu a môže sa opraviť, alebo počkať na nový príklad, ktorý sa zobrazí po niekoľkých sekundách.

Malým nedostatkom je, že v úlohe, kde treba ťahať čísla do prázdnych políčok, je možné položiť viac čísel do rovnakého políčka.



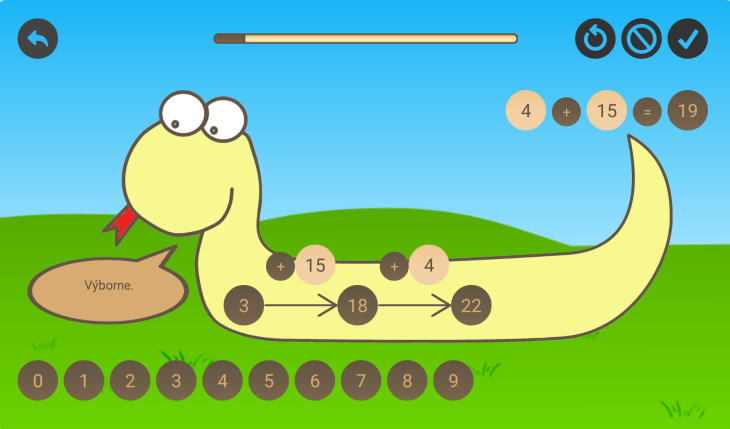
*Obrázok 9: Násobilkové štvorce*

### Hadíky

Bakalárska práca [9] vytvorená v roku 2017. Autorkou je Katarína Fabianová a spracúva prostredie Hadíky, ktoré pomáha naučiť deti rozpoznávať vzťahy medzi operátormi a operandmi. V najľahších úlohách sa počíta iba s operáciou sčítavanie, respekt. odčítavanie. Neskôr sa pridáva aj násobenie a delenie. To, s akou operáciou pracujeme je uvedené nad šípkou. Šípka vpravo potom vyjadruje sčítavanie alebo násobenie a šípka vľavo vyjadruje odčítavanie alebo delenie. Napokon sa pridá aj podmienka, čo je rovnica o dvoch neznámych.

Hra zaujme hneď na prvý pohľad, pretože je pekne graficky spracovaná, adekvátne k veku cieľovej skupiny hráčov. Ani tu nie sú žiadne prebytočné elementy, len potrebné prvky na ovládanie hry. To, či sme úlohu vyriešili správne, nám prezradí samotný had, ktorého telo je akousi hracou plochou. Pri jeho hlave sa totižto objaví bublina s textom. To je dobrá spätná väzba. Hráč si navyše môže zvoliť farbu hada.

Z pohľadu používateľa by sme ocenili, ak by sa tlačidlo na označenie úlohy, ktorá nemá riešenie a tlačidlo, ktorým označujeme, že naše riešenie je hotové a chceme vedieť, či je správne, nachádzali skôr na spodku obrazovky, nie v hornom rohu. Je totiž prirodzenejšie postupovať zhora nadol a po vyriešení úlohy hľadať tieto tlačidlá pod ňou alebo vedľa nej.



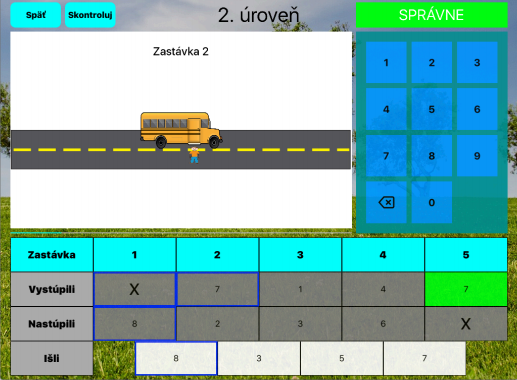
*Obrázok 10: Hadíky*

### Autobus

Bakalárska práca [10] vytvorená v roku 2017. Jej autorom je Adam Turňa. Táto mobilná aplikácia je softvérovou podporou pre prostredie Autobus. Pri úlohách tohto prostredia si deti musia veľa pamätať a pribežne sčítavať a odčítavať. S vyšším stupňom náročnosti pribúdajú políčka, do ktorých treba vpisovať správny počet cestujúcich, ktorí vystúpili, nastúpili alebo sa viezli v autobuse.

Aplikácia obsahuje všetky potrebné prvky a je príjemné s ňou pracovať. Úlohy sú zaujímavé a baví nás gradovanie. Naozaj pekným prvkom je animácia, ktorá sa spustí po správnom vyriešení úlohy. Hráč si môže ešte raz vizuálne zopakovať svoje riešenie a utvrdiť sa v ňom. Dá sa tiež chápať ako akýsi druh odmeny alebo pochvaly za dobre vykonanú prácu. V prípade, že úlohu nevyriešime správne alebo úplne, môžeme svoje riešenie opraviť.

V tejto aplikácii sa však žiada popracovať na grafickom spracovaní. Farby výberu úrovní v hlavnom menu sú príliš kontrastné s pozadím, ale naopak, farba 4. úrovne s ním splýva. Na hlavnej obrazovke hracej plochy je na prvý pohľad akosi priveľa objektov a chvíľku trvá, kým sa zorientujeme. Samotné prostredie si vyžaduje veľa komponentov a je jasné, že ich nemožno odstrániť alebo vynechať. Možno by ale bolo vhodné zriecť sa obrázku na pozadí a nahradiť ho jednofarebným pozadím.



*Obrázok 11: Autobus*

## Použité technológie

V tejto podkapitole v krátkosti priblížime technológie, ktoré budeme využívať pri tvorbe nášho softvéru.

### Android

*Android* je platforma zahŕňajúca nielen operačný systém ako taký, ale pokrýva aj používateľské rozhranie a aplikácie. O tom, že sa neustále vyvíja a zlepšuje, svedčí aj doposiaľ 17 existujúcich verzií rozšírených po celom svete. Spomedzi ostatných operačných systémov má najväčšie percentuálne zastúpenie v počte mobilných zariadení, ktoré ho používajú. Predpokladáme teda, že ak si za cieľovú platformu zvolíme práve Android, našu aplikáciu sprístupníme čo najväčšiemu možnému kruhu používateľov.

### Android Studio

*Android Studio* je vývojové prostredie určené špeciálne pre natívnu tvorbu mobilných aplikácii. Ako názov vypovedá, ide o mobilné aplikácie určené práve pre platformu Android. V súčasnosti nahrádza ešte donedávna používané Eclipse Android Development Tools. Z ponuky výhod, pre ktoré sme si ho zvolili môžeme spomenúť bohatý nástroj na tvorbu dizajnu aplikácie, či emulátor zariadení.

### Java

*Java* je objektovo orientovaný programovací jazyk vychádzajúci z jazykov C a C++. Okrem iného je to jazyk, ktorý sa využíva pri tvorbe aplikácií v Android Studiu. Jeho sila a možnosti sú tu navyše podporované ďalšími knižnicami.

### XML

XML je jednoduchý a flexibilný značkovací jazyk používaný zväčša pre prenos údajov a dokumentov, najmä na internete. Pri tvorbe našej aplikácie ho však budeme využívať na vytváranie a prispôsobovanie dizajnu podľa našich predstáv. Umožňuje nám vytvárať rôzne komponenty, napr. tlačidlá, textové polia, ktorým môžeme individuálne meniť ich vzhľad a iné parametre.

# Návrh riešenia

V tejto kapitole si rozčleníme úlohy do štyroch úrovní, oboznámime sa s jednotlivými obrazovkami aplikácie a popíšeme priebeh hry. Na záver sa pozrieme na diagram zobrazujúci pohyb používateľa v aplikácii.

## Rozdelenie úloh do úrovní

Našu aplikáciu sme rozdelili do štyroch úrovní, pričom na začiatku je odomknutá len prvá úroveň. V každej úrovni je desať úloh, ktoré je nevyhnutné vyriešiť pre možnosť vytvoriť vlastnú hru. Pre odomknutie nasledujúcej úrovne je potrebné vyriešiť prvých päť úloh.

Aby úlohy neboli jednotvárne, alebo príliš jednoduché, či naopak, príliš zložité, zadania úloh budeme gradovať v dvoch smeroch – horizontálnom a vertikálnom. Horizontálne gradovanie súvisí s rozdelením úloh do jednotlivých úrovní, ktoré detailne popíšeme v nasledujúcich podkapitolách. Vertikálne gradovanie znamená, že v rámci každej úrovne sa zadania úloh stupňujú. Prvá polovica úloh je zvyčajne triviálna, dokonca sa môže javiť až príliš ľahká. Chceme, aby žiaci na týchto úlohách pochopili zadanie, našli správne stratégie a potom sa pustili do úloh, ktoré vyžadujú väčšie úsilie. S postupným a nebadaným nárastom si žiaci neuvedomujú, že druhá polovica úloh je naozaj náročná. Navyše, aj slabší žiaci sú schopní vyriešiť časť úloh a to vnímame ako veľkú motiváciu pre nich samotných.

### Prvá úroveň

Úlohy v prvej úrovni si vyžadujú nájsť jedno správne riešenie. Všetky parkety musia byť použité, žiadna nie je nazvyš. Parkety sú rozmiestnené osobitne, na každú z nich hráč vidí. Tento typ úlohy je najjednoduchší, pretože hráč si je istý, že riešenie existuje a nemusí rozhodovať o tom, ktorý tvar parkiet použije, alebo nepoužije. Hoci úloha môže mať veľa riešení, stačí ak nájde práve jedno z nich.

### Druhá úroveň

Do úloh z tejto úrovne pribudli aj parkety nazvyš. Teraz sú všetky parkety rovnakého tvaru položené na jednej kope a ich počet označuje číslo. Náročnosť oproti prvej úrovni je vyššia, pretože niekedy je potrebné použiť parkety z viacerých ponúkaných typov, inokedy stačí použiť jeden typ parkiet. Hoci riešení môže byť viac, na úspešné zvládnutie úlohy je stále potrebné nájsť len jedno z nich.

### Tretia úroveň

Tretia úroveň prináša oproti predchádzajúcej hneď dve zmeny. Prvou z nich je preddefinovaná parketa na podlahe, s ktorou hráč nemôže hýbať. Jednoducho ukladá parkety na zvyšné voľné miesto. Druhou zmenou je, že teraz sa od neho očakáva nájdenie všetkých možných riešení. Náročnosť tejto úlohy podporuje aj fakt, že vopred nevie, koľko riešení úloha má. Vyberáme však také úlohy, ktoré majú najmenej jedno a najviac šesť možných riešení. Riešenia, ktoré sú len otočené o 90 stupňov voči prechádzajúcim, nepovažujeme za nové.

### Štvrtá úroveň

Na rozdiel od predchádzajúcich, posledná úroveň obsahuje aj úlohy, ktoré nemajú riešenie. Stále však platí, že sa v nej nachádzajú aj úlohy, ktoré majú viac riešení a v tom prípade je potrebné nájsť všetky. Podlaha neobsahuje žiadnu preddefinovanú parketu, takže počet možností uloženia parkiet je vyšší. Teraz musí hráč rozhodnúť, či úloha má alebo nemá riešenie. Ak áno, musí nájsť všetky riešenia a navyše, niektoré parkety pri ich hľadaní nemusí vôbec použiť. Takéto zadanie je naozaj náročné.

## Úvodná obrazovka

Po spustení aplikácie sa používateľovi zobrazí úvodná obrazovka, ktorú môžeme nazvať aj hlavné menu. To pozostáva z piatich tlačidiel. Cez tlačidlo s ikonou ozubeného kolieska sa dostaneme do obrazovky nastavení. Ostatné tlačidlá spúšťajú jednotlivé úrovne. Ak je úroveň odomknutá, na tlačidle je zobrazené číslo. V opačnom prípade je na tlačidle ikona zámok, čo signalizuje zatiaľ neprístupnú úroveň. Zároveň sme každú úroveň farebne odlíšili a toto farebné odlíšenie uvidíme aj v nasledujúcich obrazovkách. Každá z obrazoviek totiž mení farbu niektorého komponentu práve podľa úrovne, v ktorej sa práve nachádzame.

Obrázok, na ktorom je vektorová grafika

Popis vygenerovaný s vysokou spoľahlivosťou

*Obrázok 12: Úvodná obrazovka*

## Hlavná obrazovka s úlohami

Kľúčová obrazovka našej aplikácie je jednoznačne hlavná obrazovka s úlohami. Samotné zadanie hry, teda podlaha a pakety, obsahuje veľa objektov, ktoré je potrebné vykresliť. Preto nechceme plochu zbytočne zahltiť ďalšími komponentami. Obrazovka teda obsahuje len samotnú hraciu plochu a vrchný panel, ktorý obsahuje komponent pre zobrazovanie poradia úlohy zo základu desať (ďalej progressbar). Ak hráč úspešne vyriešiť päť úloh, odomkne sa ďalšia úroveň. Zároveň sa vo vrchnom paneli vedľa progressbaru zobrazí ikona odomknutej úrovne. To logicky neplatí pre poslednú úroveň.

### Priebeh hry

Po zvolení úrovne v menu sa hráčovi vykreslí hracia plocha. Tomu však predchádza krátka animácia zobrazujúca chlapca rozmýšľajúceho nad novým zadaním. V skutočnosti však na pozadí prebieha generovanie novej úlohy, ktoré môže chvíľu trvať.

Parkety sú v prvej úrovni rozmiestnené jednotlivo (obr. 13a), v ostatných úrovniach sú vykreslené pre každý typ parkety len raz a ich počet predstavuje príslušné číslo (obr. 13b). Číslo neskôr predstavuje počet parkiet, ktoré ešte nie sú na podlahe, teda sa počas manipulácie s parketami mení. Hráč parkety presúva po ploche ťahaním prstom. Otočenie parkety o 90 stupňov doprava je možné dvojitým kliknutím na príslušnú parketu.

V prvej a druhej úrovni stačí nájsť jedno riešenie. Po jeho nájdení sa okraje plochy zafarbia na zeleno a zobrazí sa tlačidlo so šípkou (obr. 13c). Po kliknutí na toto tlačidlo sa opakuje proces prípravy novej hry a zároveň sa zmení stav progressbaru.

V tretej a štvrtej úrovni je potrebné nájsť všetky riešenia. Ak hráč správne pokryje podlahu, jej okraje sa zafarbia na zeleno. Hráč môže pokračovať v hľadaní riešenia, alebo môže kliknúť na tlačidlo, ktoré zavolá novú obrazovku, ktorú detailne popisujeme v kapitole 2.4. V prípade, že počet riešení je správny, táto obrazovka sa automaticky zatvorí, zmení sa stav progressbaru a sa spustí príprava ďalšej úlohy. Inak sa hráč kliknutím na ľubovoľné miesto vráti na hraciu plochu a ďalej hľadá riešenia. Aj nájdenie každého ďalšieho riešenia signalizuje zelený okraj podlahy. Ak hráč správne pokryl podlahu, ale toto riešenie sa mu podarilo nájsť už pred tým, alebo ak je toto riešenie len otočením nejakého predchádzajúceho riešenia, okraje podlahy sa zafarbia na oranžovo (obr. 13d). Riešenie je teda správne, ale nie nové.

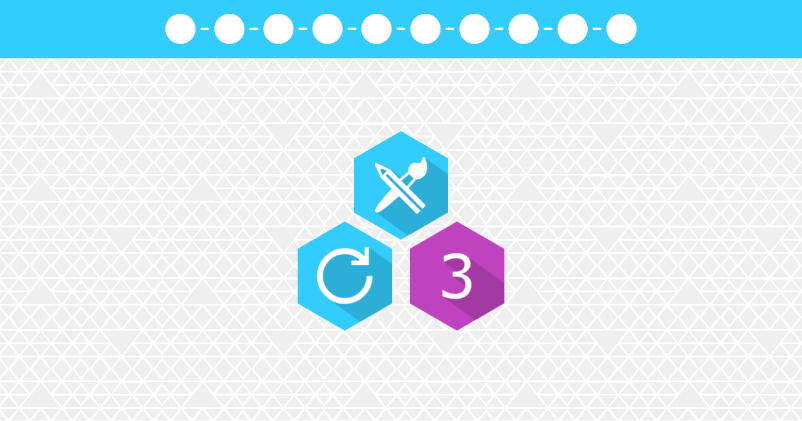
Obrázok, na ktorom je krížovka

Popis vygenerovaný s veľmi vysokou spoľahlivosťou

*Obrázok 13: Hlavná obrazovka s úlohami pre rôzne herné scenáre*

### Obrazovka po dokončení úrovne

Ak hráč úspešne zvládol desať úloh, má tri možnosti. Môže si vypýtať ďalšiu úlohu kliknutím na tlačidlo Opakuj. Vygeneruje sa nová úloha s náročnosťou rovnakou ako má úloha desať. Kliknutím na tlačidlo s ikonou štetca si môže vytvoriť vlastnú úlohu. Napokon, kliknutím na tlačidlo s ikonou ďalšej úrovne môže začať riešiť úlohy danej úrovne. Na obrázku 14 je obrazovka po dokončení druhej úrovne.



*Obrázok 14: Obrazovka po dokončení celej úrovne*

# Realizácia riešenia

V tejto kapitole si priblížime samotnú implementáciu aplikácie. Najskôr sa oboznámime s jej štruktúrou, spomenieme niektoré zaujímavé triedy objektov, alebo čiastkové problémy, s ktorými sme sa stretli a vyžadovali si náročnejšie riešenie.

## Štruktúra aplikácie

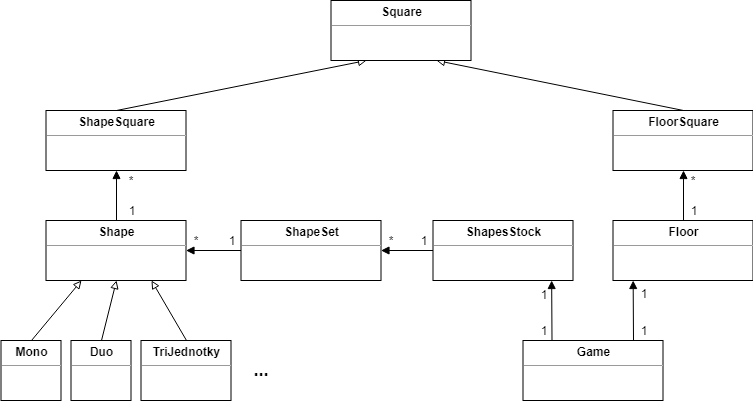
Pri implementovaní sme vytvorili niekoľko balíčkov. Triedy objektov sú v nich zoskupené podľa funkcionality, ktorú nám poskytujú. V krátkosti si ich popíšeme a na niektoré z nich sa bližšie pozrieme v nasledujúcich kapitolách.

* **Activities**

Obsahuje tzv. aktivity, čo sú základné komponenty Android aplikácie. Tie sú prepojené a dokopy tvoria aplikačný model. O pohybe používateľa medzi týmito aktivitami sme hovorili v kapitole 2.7 Mapa pohybu používateľa v aplikácii. Veľmi zjednodušene môžeme povedať, že sú to triedy, ktoré sa starajú o vykreslenie a správanie jednotlivých obrazoviek.

* **Game**

Nachádzajú sa v ňom triedy, ktoré súvisia so samotnou úlohou. Teda podlaha (*Floor*) a set parkiet (*ShapesStock*). Podlahu tvorí kolekcia jednotlivých štvorcov (*FloorSquare*). Sada parkiet zase obsahuje akési množiny (*ShapeSet*). Tieto množiny si pamätajú tvar parkety, počet kusov a informáciu o tom, či musí byť každá parketa použitá práve raz, alebo nemusí byť použitá vôbec. Tiež obsahuje samotný zoznam konkrétnych inštancií triedy pre parkety (*Shape*). *Shape* je nadtrieda pre parkety, pričom každý tvar má vlastnú podtriedu s názvom zaužívaným v školskom prostredí, napr. *Mono*, *Duo*, *TriJednotky* a pod. Jeden objekt triedy Shape obsahuje dvojrozmerné pole tvorené objektami triedy *ShapeSquare*. Zadanie jednej úlohy (*Game*) sa skladá a z *Floor* a *ShapesStock*. Zadanie úlohy, ktorá pochádza od používateľa a bola vytvorená v editore má vlastnú triedu (*GameFromEditor*), pretože má odlišné atribúty a funkcie, ktoré potrebujeme najmä na hľadanie riešení a na následnú konverziu. Pre lepšiu orientáciu slúži obrázok 12. Ten veľmi zjednodušene zobrazuje tieto triedy a vzťahy medzi nimi. Niektoré úlohy vyžadujú nájdenie všetkých riešení. Tieto riešenia si teda musíme pamätať a pracovať s nimi. Na to nám slúži trieda riešenie (*SolutionsStock*), ktorá má podobnú štruktúru – obsahuje zoznam všetkých riešení, čo sú objekty triedy *Solution*. Tieto sú zase tvorené malými štvorcami reprezentovanými triedou *SolutionSquare*. *FloorSquare*, *SolutionSquare* a *ShapeSquare* sú podtriedami triedy *Square*.



*Obrázok 15: Diagram zobrazujúci vzťahy medzi niektorými triedami*

* **Services**

Services je balíček obsahujúci prevažne statické triedy. Teda inštancie týchto tried nemusíme vytvárať, ak chceme využívať ich funkcionalitu. Z týchto spomenieme napríklad *Generator* a *Solver*. Statická trieda *Shared* obsahuje niekoľko konštánt a funkcií, ktoré sú volané a využívané z najrôznejších tried. To nám umožňuje vyhnúť sa duplicite kódu. Napríklad pri neskoršej zmene niektorého parametra tak stačí túto zmenu vykonať raz s garanciou, že všetky objekty budú pristupovať k novej hodnote. Ďalej je to trieda *InitialGrid*, ktorá sa stará o počiatočné rozmiestnenie parkiet, teda priraďuje parketám ich súradnice. Trieda *GameController* nie je statická a jej inštancia sa vytvára vždy pri spustení hlavnej obrazovky s hracou plochou. Stará sa o stav hry, pamätá si predchádzajúce zadanie úlohy, ale aj doteraz nájdené riešenia. Na požiadanie zavolá niektoré z vyššie uvedených servisných tried, napríklad aby vygenerovala nové zadanie úlohy. Obsahuje aj triedu *State*, v ktorej sú uložené údaje o tom, ktoré úrovne sú odomknuté a koľko úloh v rámci nich už používateľ úspešne zvládol. Tieto údaje sa vždy pri spustení aplikácie načítajú z pamäte a pri ukončení aplikácie sa do nej uložia.

* **Resources**

Automaticky vytvorený balíček, ktorý obsahuje obrázky, súbory na vykresľovanie grafických komponentov, ale aj zadefinované hodnoty a iné. My využívame najmä podadresár *Layout*, kde sú xml súbory pre jednotlivé okná aplikácie, či *Drawable*, kde sú uložené ikony a spomínané obrázky.

* **SolverDLX**

Balíček *SolverDLX* obsahuje triedy zabezpečujúce hľadanie riešení pre problém pokrytia. Jednak sú to triedy reprezentujúce našu triedu *Game*, ale aj triedy, ktorých úlohou je konvertovať zadanie úlohy do správneho vstupu pre ďalšie prehľadávanie.

## Parkety

Kľúčovými objektami sú pre nás práve parkety. Celkový dojem z aplikácie totiž z veľkej časti záleží práve na ich pohodlnom ovládaní. Každá parketa obsahuje atribúty pre farbu, aktuálnu a počiatočnú pozíciu na vodorovnej a zvislej osi, identifikačné číslo a atribúty pre detekciu a prácu pri dotyku. Spomedzi funkcií je zaujímavá napríklad funkcia pre otočenie matice o 90 stupňov v smere hodinových ručičiek, keďže jednotlivé štvorce, ktoré tvoria parketu, máme uložené ako dvojrozmerné pole. Parketu môžeme otočiť dvojitým kliknutím, ktoré jednoducho detegujeme pomocou *GestureListener*-a. Ostatné funkcie prevažne zisťujú, či sa parketa nachádza na podlahe alebo mimo hracej plochy. To všetko súvisí aj s presúvaním parkiet po hracej ploche. Z hľadiska implementácie to nie je úplne triviálny problém, preto mu venujeme nasledujúcu podkapitolu.

### Presúvanie parkiet po hracej ploche

Zo zachyteného dotyku na obrazovke zariadenia dostávame jeho presné súradnice. Ak bol dotyk zaznamenaný, prejdeme každú parketu v našej kolekcii a zistíme, či bola zasiahnutá. A to tak, že pre každý štvorec, ktorý ju tvorí, vypočítame jeho reálne súradnice na ploche a porovnáme ich so súradnicami dotyku. Rátame aj s malou odchýlkou, ktorá je potrebná najmä pri zariadeniach s menšími obrazovkami.

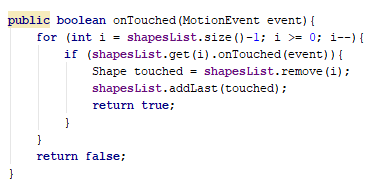
Okrem toho sme zabezpečili, že:

* *Parketa sa nedostane za okraje plochy*

Ošetrili sme situáciu, kedy by sa parketa nechceným pohybom dostala za okraje plochy, čo by znemožnilo riešenie úlohy.

* *Presúvaná parketa sa vždy nachádza nad ostatnými parketami*

Pre používateľa je prirodzené, že objekt, s ktorým manipuluje, sa nachádza úplne navrchu a má naň neustály dohľad. Želaným správaním je tiež pohybovanie len jednou parketou, nie viacerými naraz. To sa nám podarilo zabezpečiť jednoduchým trikom. Všetky parkety sú v spájanom zozname. Ten pri dotyku prechádzame od konca a ak bola niektorá parketa dotknutá, zaradíme ju na koniec zoznamu a ďalej už nepokračujeme. Keďže vykresľovanie parkiet prebieha od začiatku zoznamu, hýbaná parketa sa vykreslí ako posledná nad všetkými ostatnými. Zároveň sa tým nedeteguje dotyk parkiet, ktoré sú vykreslené pod ňou a nedochádza k pohybu viacerých parkiet naraz, ak sa prekrývajú.



*Obrázok 16: Riešenie problému posúvania parkiet po ploche*

* *Parketa automaticky dosadne do podlahy*

Ak sa parketa po pustení nachádza na podlahe, vypočítame, na ktoré miesto ju používateľ pravdepodobne chcel položiť a zmeníme jej súradnice. Vytvára to dojem „cvaknutia“. Tým tiež zabezpečíme, že parkety presne kopírujú podlahu a nevytŕčajú z nej. Toto automatické dosadenie sa však udeje len vtedy, ak dané štvorce podlahy ešte nie sú obsadené inými objektami.

* *Parketa sa vráti na svoje pôvodné miesto*

Takýto jav nastane, ak bola parketa na podlahe a používateľ ju z nej dlhším ťahom presunie na pravú časť hracej plochy.

## Generátor úloh

Keďže úlohy nie sú napevno zadané a ich počet nie je limitovaný, využívame generátor úloh. Celé úrovne sa tak dajú hrať aj opakovane a vždy budú odlišné a vyžadujúce hľadanie nových riešení. Zadanie úlohy sa skladá z dvoch častí – podlaha a parkety. Pri vytváraní nových úloh sa najskôr vygeneruje štvorcová sieť. Ak ide o úroveň, v ktorej sa na podlahe nachádza preddefinovaná parketa, náhodne sa zvolí jej tvar a umiestnenie. Hoci chceme, aby bol generátor náhodný, je to tak len do istej miery. Stupňovanie náročnosti súvisí aj s rozmermi plochy. Na začiatku sú to dva riadky a dva stĺpce, postupne sa zväčšujú v oboch smeroch. Generovanie parkiet tiež nie je úplne náhodné. V prvej polovici úrovne sa stretneme s jednoduchými tvarmi ako sú napr. Mono a Duo. Umiestňovanie týchto parkiet je jednoduché a naviac, iné parkety by sa zatiaľ do plochy nezmestili. V druhej polovici sa však generujú najmä zložitejšie tvary a tie jednoduchšie sa pridajú, ak treba vyplniť malú zostatkovú plochu.

Takto vygenerovaná úloha sa pošle na overenie do *Solvera*, ktorý určí, či má úloha správny počet riešení. Ak áno, zadanie úlohy je hotové, parketám sa priradia farby a počiatočné umiestnenie na ploche. Ak nie, proces generovania parkiet sa zopakuje.

## Solver

*Solver* je statická trieda, ktorú využívame na hľadanie počtu riešení úlohy. Jej funkcia ako parameter dostáva objekt triedy Game a číslo označujúce hornú hranicu počtu riešení. Úlohy, ktoré obsahujú podlahu veľkých rozmerov (napr. 5x5) a ku ktorej logicky prislúcha aj väčší počet parkiet (napr. 10), majú často viac než stovku riešení. V úrovniach, kde požadujeme nájdenie všetkých riešení pre nás takáto úloha nie je vyhovujúca. Ak je teda počet nájdených riešení len o jedno väčší ako zadaná horná hranica, hľadanie ďalších už pre nás nemá zmysel a zastaví sa. Okrem iného sa tým aj urýchli doba čakania.

Keďže hľadáme všetky možné pokrytia podlahy parketami, vzniká tu zložitý kombinatorický problém, ktorý môžeme previesť do problému *Exact cover*. Na jeho riešenie sa používa technika *Dancing links*, ktorej existujúcu implementáciu [11] sme čiastočne upravili tak, aby vedela riešiť práve náš problém s parketami a podlahou. Samotná trieda *Solver* je teda len komunikátorom medzi našimi objektami a skutočným rozsiahlejším balíčkom *SolverDLX* na hľadanie riešení. Najskôr sa objekty triedy *Floor* a *ShapesStock* jednoduchými funkciami prekonvertujú na objekty triedy *Puzzle* a *ParquetSet*. Následne sa zavolá funkcia, ktorá ako parametre očakáva práve tieto nové objekty a číslo označujúce hornú hranicu počtu riešení. Funkcia ako výsledok vráti číslo menšie alebo rovné vstupnému parametru. Ak je menšie, znamená to, že úloha má presne toľko riešení. Ak je rovné, znamená to, že úloha má najmenej toľko riešení. S týmto výsledkom ďalej pracujeme v závislosti od úrovne a rozhodujeme sa, či je to pre nás dostačujúce, nedostačujúce alebo nadmerné množstvo.

Na spomínaný problém a techniku jeho riešenia sa bližšie pozrieme v nasledujúcich dvoch podkapitolách. Na záver aplikujeme tieto poznatky na riešenie nášho problému.

### Exact cover problem

*Exact cover problem*, čo môžeme voľne preložiť ako *problém presného pokrytia*, je NP‑ťažký problém.

Definícia: Nech ***S*** je systém množín, ktorého prvkami sú podmnožiny množiny *X*. Problém presného pokrytia je podmnožina ***S\**** z ***S*** taká, že každý prvok z *X* sa nachádza práve v jednej z podmnožín ***S\****.

Problém môžeme ilustrovať na jednoduchom príklade:

Nech ***S*** = {A, B, C, D} je súbor podmnožín *X* = {1, 2, 3, 4, 5} takých, že:

A = {1, 2};

B = {2, 3};

C = {1, 2, 3};

D = {4, 5}.

Hľadaná podmnožina ***S\**** je teda {C, D} pretože sa každý prvok z množiny *X* nachádza v niektorej z jej prvkov práve raz. C a D sú disjunktné (t.j. ich prienikom je prázdna množina) a ich zjednotenie je *X*. Iné podmnožiny ***S\**** nie sú presným pokrytím *X*, pretože buď nie sú navzájom disjunktné, alebo nepokryjú všetky prvky *X*.

Častým spôsobom reprezentácie tohto problému je aj matica výskytu, ktorá má rozmery . Každý riadok matice predstavuje jeden prvok z ***S*** a každý stĺpec predstavuje jeden prvok z *X*. Hodnota prvku matice je 1, ak sa prvok korešpondujúceho stĺpca nachádza v množine korešpondujúceho riadku. Inak je hodnota prvku matice 0. Presné pokrytie je potom výber takých riadkov, že každý stĺpec obsahuje 1 práve v jednom zo zvolených riadkov. Matica výskytu pre náš ilustračný príklad je zobrazená na obrázku 14.

# 

*Obrázok 17: Matica výskytu pre problém presného pokrytia*

Problém pokrytia je teda rozhodovací problém, či takéto pokrytie existuje. Nájdeme ho napríklad v Sudoku alebo Pentomino puzzle, či dokonca v mierne zovšeobecnenej verzii aj v probléme N dám.

### Dancing links

*Dancing links* je technika navrhnutá Donaldom E. Knuthom pre implementovanie jeho *Algoritmu X*. *Algoritmus X* je rekurzívny nedeterministický algoritmus na hľadanie všetkých riešení pre problém presného pokrytia. Keďže matica výskytu má pre niektoré problémy obrovské rozmery, nájdenie výskytu 1 v nej môže trvať príliš dlho.

V technike *Dancing links* sa riadky a stĺpce matice za účelom nájdenia výskytu 1 neprechádzajú celé. Pre každý stĺpec je vytvorený špeciálny vrchol, z ktorého môžeme v lineárnom čase zistiť celkový počet 1 v danom stĺpci. Vrcholy, ktoré obsahujú 1 sú totiž prepojené dvojitým spájaným zoznamom a to so susedmi zľava a sprava v rovnakom riadku a so susedmi zhora a zdola v rovnakom stĺpci. Práve z toho vzniklo pomenovanie tejto techniky, keďže vrcholy pripomínajú tanečníkov. Ďalej sa potom pracuje s touto štruktúrou, pri ktorej sa časová zložitosť prehľadávania z O(n) zmenšila na O(1). Knuth opisuje implementáciu *Dancing links* podrobnejšie vo svojom článku [12].

### Exact cover problem v našej aplikácii

Aby sme mohli využívať existujúci algoritmus na riešenie problému pokrytia aj v našej aplikácii, musíme previesť zadanie úlohy do vhodného formátu vstupu. Vytvoríme teda maticu výskytu podľa nasledujúcich pravidiel: Počet stĺpcov matice sa rovná súčtu voľných štvorcov na podlahe a počtu parkiet. Pre každú parketu zistíme koľko je možností jej uloženia na plochu, počítame aj s otočeniami. Počet riadkov matice je potom súčet všetkých týchto možností uloženia na plochu pre každú parketu. Napokon pre každú parketu a každé jej otočenie zvolíme umiestnenie na podlahe. To predstavuje 1 v matici. Obrázok 15 zobrazuje maticu výskytu pre jednoduchú úlohu, kde podlaha má rozmery 2×2 a kolekcia parkiet obsahuje dvakrát Mono a raz Duo. Takáto matica sa potom prevedie na obojsmerný spájaný zoznam podľa techniky *Dancing links*. Na tomto spájanom zozname sa následne spustí algoritmus prehľadávania, ktorý zistí, či existuje úplné pokrytie. Ak áno, naša úloha má riešenie, ak nie, úlohu nie je možné vyriešiť. S malými vylepšeniami táto implementácia funguje aj pre úlohy, v ktorých nemusia byť použité všetky parkety. Ak je možné nájsť viac pokrytí, znamená to, že úloha má viac riešení. Pozor, v tomto prípade sa aj otočenie o 90 stupňov chápe ako nové riešenie!

# 

*Obrázok 18: Matica výskytu pre problém presného pokrytia jednej z úloh*

## Responzívnosť

Responzívna aplikácia je taká aplikácia, ktorá sa vie prispôsobiť obrazovke zariadenia, na ktorom je spustená. Keďže chceme aby bola naša aplikácia prístupná čo najväčšiemu počtu žiakov, musíme zabezpečiť, aby bola použiteľná nezávisle od zariadenia. Hoci sa na niektorých zariadeniach môžu niektoré komponenty zobrazovať trochu rozdielne, myslíme si, že tieto rozdiely nemajú dopad na hrateľnosť a ovládanie.

Z hľadiska dizajnu sa nám to podarilo vyriešiť najmä vďaka vstavaným nástrojom v Android Studiu, z ktorých spomenieme napríklad vodiace čiary. Ich polohu nastavujeme v percentách a nie v pixeloch. Pred vykresľovaní podlahy a parkiet sa najskôr vypočíta veľkosť hrany podľa informácie o veľkosti obrazovky daného zariadenia. Rovnakým spôsobom ošetrujeme aj to, aby parkety nemohli prejsť mimo hracej plochy. Aplikácia je okrem toho vždy zobrazovaná na šírku, čím máme zaistené primerané rozmiestnenie objektov. Na obrázku 16 je aplikácia spustená na telefóne s 5 palcovým displejom a rozlíšením 1920×1080 pixelov a na tablete s 9,94 palcovým displejom a rozlíšením 2560×1800 pixelov.

Obrázok, na ktorom je snímka obrazovky

Popis vygenerovaný s veľmi vysokou spoľahlivosťou

*Obrázok 19: Aplikácia bežiaca na zariadeniach s rozdielnym rozlíšením*

# Zdroje

[1] LEHOTSKÁ, D., Edukačný softvér, 2013. [online]. Dostupné na internete:

<<http://files.virtual-lab.sk/APVV%20VV-11/Clanky%20a%20publikacie/Edukacny%20softver_Daniela_Lehotska_Bezakova.pdf>>

[2] Edukačný softvér [online]. [2.1.2018]. Dostupné na internete:

<http://www.zspodzavozca.edu.sk/ulohy/edukacnysoftver.pdf>

[3] HEJNÝ, M. – JIROTKOVÁ, D. – SLEZÁKOVÁ-KRATOCHVÍLOVÁ, J. 2015. Matematika pre 2.ročník – Pracovná učebnica. Bratislava: Indícia, 2015. 64 s.

978-80-971734-3-2

[4] Magnetické parkety z prostředí prof. Hejného [online]. [4.1.2018]. Dostupné na internete: <<http://www.hrouzavylohou.cz/vylep-na-nastenku/nase-magneticke-parkety-z-prostredi-pro-hejneho>>

[5] Náhradní dílky k dřevěným parketám [online]. [4.1.2018]. Dostupné na internete:

<<http://www.h-ucebnice.cz/product/didakticke-pomucky/nahradni-dilky-k-drevenym-parketam/24>>

[6] Úlohy z matematiky pre deti na základných školách [online]. [11.1.2018]. Dostupné na intrernete: <<https://www.matika.in/sk/>>

[7] Matemág – Dobrodružná edukativní hra CZ, TechSophia [online]. [11.1.2018]. Dostupné na internete:

<<https://play.google.com/store/apps/details?id=cz.techsophia.mathtraining.mathstory>>

[8] Martin Sadloň: Softvérová podpora vyučovania matematiky Hejného metódou – prostredie Násobilkové štvorce, bakalárska práca, FMFI UK Bratislava, 2017

[9] Katarína Fabianová: Softvérová podpora vyučovania matematiky Hejného metódou – prostredie Hadíky, bakalárska práca, FMFI UK Bratislava, 2017

[10] Adam Turňa: Softvérová podpora vyučovania matematiky Hejného metódu – prostredie Autobus, bakalárska práca, FMFI UK Bratislava, 2017

[11] https://github.com/benfowler/dancing-links

[12] KNUTH, E. Donald: *Dancing links,* Millennial *Perspectives in Computer Science, Palgrave, 187-214, 2000*. Dostupné na internete:

<http://www-csfaculty.stanford.edu/~knuth/papers/dancing-color.ps.gz>