**UNIVERZITA KOMENSKÉHO V BRATISLAVE**

**FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY**

**SOFTVÉROVÁ PODPORA VYUČOVANIA MATEMATIKY**

**HEJNÉHO METÓDOU – PROSTREDIE JEŽIBABA ARMIDA**

Bakalárska práca

**Bratislava, 2023**  Rebeka Mikulášová

**UNIVERZITA KOMENSKÉHO V BRATISLAVE**

**FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY**

**SOFTVÉROVÁ PODPORA VYUČOVANIA MATEMATIKY**

**HEJNÉHO METÓDOU – PROSTREDIE JEŽIBABA ARMIDA**

Bakalárska práca

Študijný program: Aplikovaná informatika

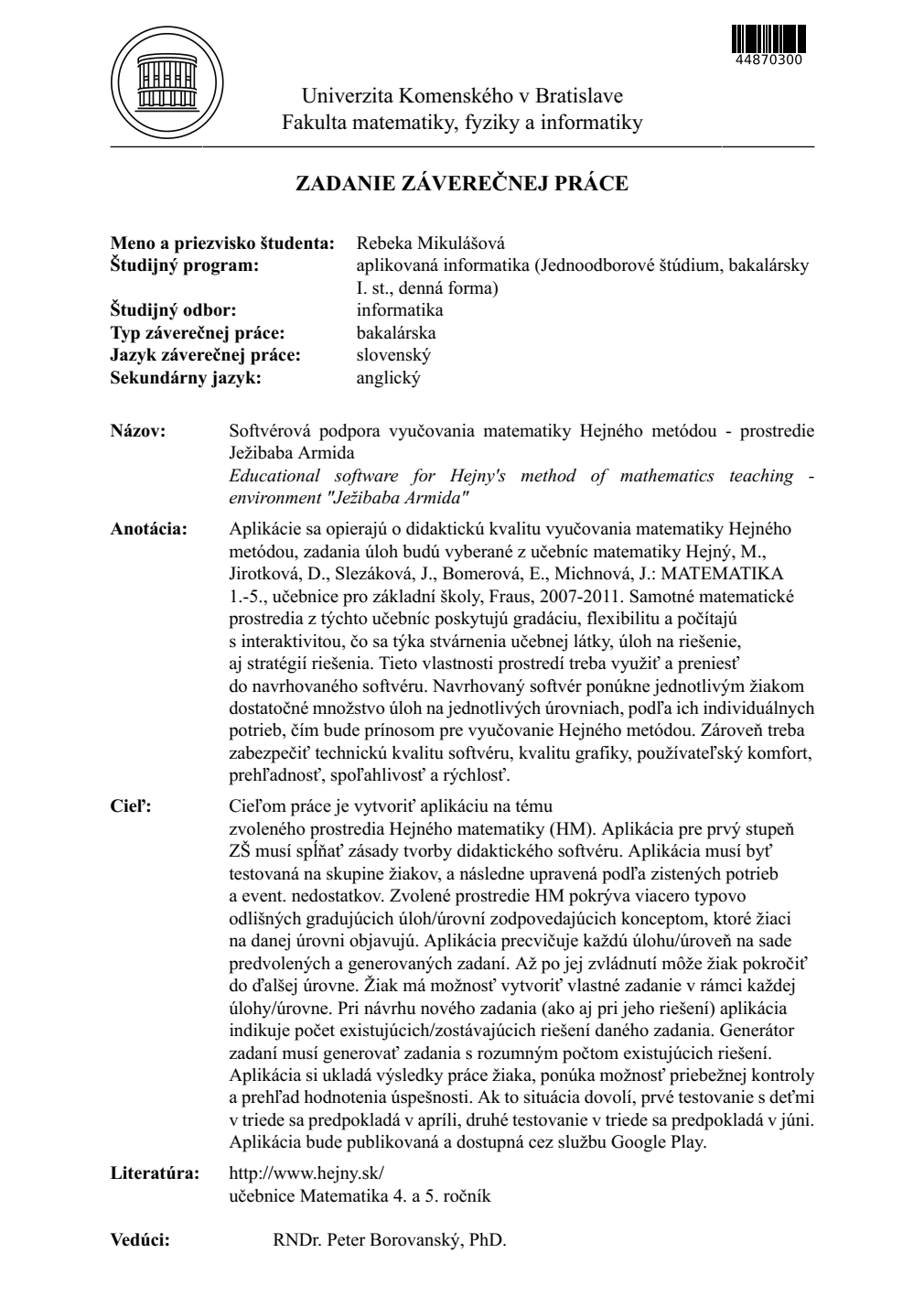
Študijný odbor: 2511 Aplikovaná informatika

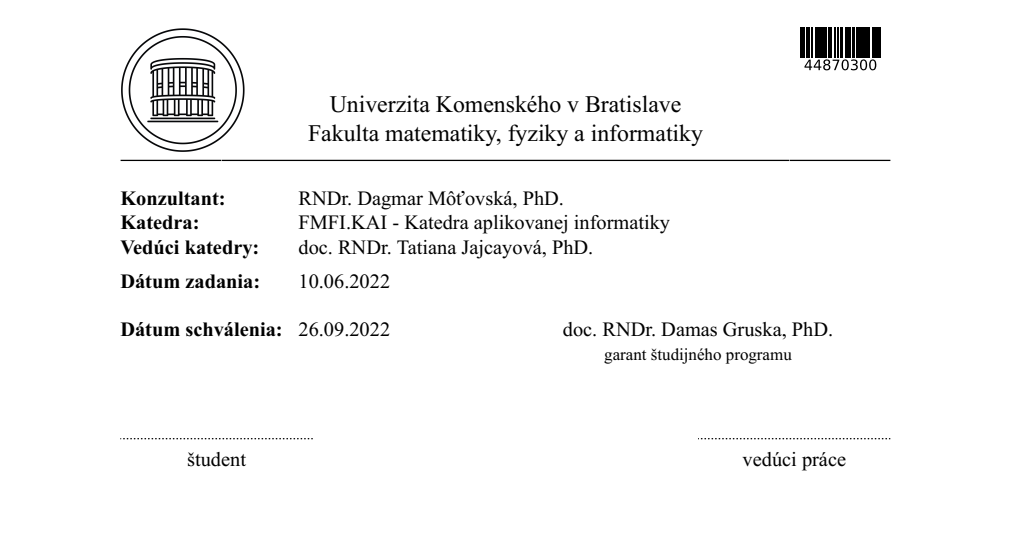
Školiace pracovisko: Katedra aplikovanej informatiky

Školiteľ: RNDr. Peter Borovanský, PhD.

Konzultant: RNDr. Dagmar Môtovská, PhD.

**Bratislava, 2023**  Rebeka Mikulášová





**Čestné prehlásenie**

Čestne prehlasujem, že som túto bakalársku prácu vypracovala samostatne pod vedením vedúceho bakalárskej práce, s použitím uvedenej literatúry a zdrojov dostupných na internete.

Bratislava, 16.5.2023

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Rebeka Mikulášová

**Poďakovanie**

**Abstrakt**

**Abstract**

# Obsah

[Obsah 9](#_Toc134726107)

[1. Východisková kapitola 14](#_Toc134726108)

[1.1 Edukačný softvér 14](#_Toc134726109)

[1.2 Hejného metóda 14](#_Toc134726110)

[1.3 Prostredie Ježibaba Armida 17](#_Toc134726111)

[1.4 Diofantické rovnice 18](#_Toc134726112)

[1.4.1 Diofantos a vznik diofantických rovníc 18](#_Toc134726113)

[1.4.2 Riešenie pomocou rozšíreného Euklidovho algoritmu 19](#_Toc134726114)

[1.4.3 Riešenie pomocou kongruencie 22](#_Toc134726115)

[1.4.4 Riešenie pomocou redukčnej metódy 24](#_Toc134726116)

[1.4.5 Diofantické rovnice o troch neznámych 24](#_Toc134726117)

[1.5 Podobné práce 24](#_Toc134726118)

[1.5.1 Math Slide 24](#_Toc134726119)

[1.5.2 Matemág 25](#_Toc134726120)

[1.5.3 Krokovanie 26](#_Toc134726121)

[2. Návrh 29](#_Toc134726122)

[2.1 Používateľské rozhranie a logika úrovní 29](#_Toc134726123)

[2.1.1 Menu 29](#_Toc134726124)

[2.1.2 Úrovne 1 a 2 30](#_Toc134726125)

[2.1.3 Úrovne 3 a 4 32](#_Toc134726126)

[2.1.4 Playground 36](#_Toc134726127)

[2.2 Editory úloh 36](#_Toc134726128)

[3 Implementácia 40](#_Toc134726129)

[3.1 Štruktúra aplikácie 40](#_Toc134726130)

[3.2 Implementácia úrovní 41](#_Toc134726131)

[3.3 Ovládanie aplikácie 43](#_Toc134726132)

[3.4 Animácie a zvuk 45](#_Toc134726133)

[4. Testovanie 46](#_Toc134726134)

[4.1 Cieľ testovania 46](#_Toc134726135)

[4.2 Priebeh testovania 46](#_Toc134726136)

[4.3 Vyhodnotenie testovania 47](#_Toc134726137)

[5. Zdroje 50](#_Toc134726138)

**Zoznam obrázkov**

Obrázok 1: Príklad úlohy z prostredia Ježibaba Armida

Obrázok 2: Príklad úlohy z prostredia Ježibaba Armida

Obrázok 3: Math Slide menu

Obrázok 4: Math Slide hra

Obrázok 5: Matemág, tretia úroveň

Obrázok 6: Matemág, komiks

Obrázok 7: Krokovanie, hlavné menu

Obrázok 8: Krokovanie, prvá úroveň

Obrázok 9: Hlavné menu s uzamknutými úrovnami

Obrázok 10: Hlavné menu po otvorení všetkých úrovní

Obrázok 11: Obrazovka prvej úrovne

Obrázok 12: Obrazovka druhej úrovne

Obrázok 13: Obrazovka tretej úrovne

Obrázok 14: Obrazovka zadaných riešení

Obrázok 15: Obrazovka druhej polovice tretej úrovne

Obrázok 16: Obrazovka štvrtej úrovne

Obrázok 17: Obrazovka playground

Obrázok 18: Editor prvej úrovne

Obrázok 19: Editor druhej úrovne

Obrázok 20: Editor tretej úrovne

Obrázok 21: Editor štvrtej úrovne

Obrázok 22: Assets folder

Obrázok 23: Prechod medzi snénami (scene flow chart)

Obrázok 24: Zabezpečenie perzistencie Game Managera

Obrázok 25: Kód triedy Loader

Obrázok 26: Metóda OnBeginDrag

Obrázok 27: Testovanie aplikácie

Obrázok 28: Testovanie aplikácie

**Úvod**

# Východisková kapitola

V tejto kapitole sa venujem významu edukačného softvéru, princípom Hejného metódy výučby matematiky a prostrediu Ježibaba Armida, prejdem aj existujúce implementácie podobných softvérov a na záver priblížim diofantické rovnice.

## Edukačný softvér

Softvér alebo program môžeme nazvať edukačným, ak je určený k výučbovým účelom a využíva didaktické funkcie. V posledných troch rokoch sme mohli vidieť význam a opodstatnenie edukačných softvérov vo vyučovacom prostredí. Deti si zvykli na riešenie úloh pomocou počítačov, tabletov a mobilných telefónov. Vďaka multimediálnemu obsahu, ako napríklad grafika, obrázky, animácie a zvuk je interaktívne riešenie zadaní je pre nich zaujímavejším a viac pútavým spôsobom učenia. Edukačný softvér [1] je benefitom aj pre učiteľov, ktorým dovoľuje lepšie spojenie so žiakmi a pomáha im zvýšiť u žiakov záujem o učenie.

Pri tvorbe edukačného softvéru musíme myslieť na jeho základné vlastnosti:

* Intuitívne, ľahko ovládateľné používateľské prostredie
* Interaktívnosť
* Možnosť užívateľa tvoriť nový materiál
* Vhodná gradácia náročnosti úrovní a úloh
* Vhodnosť programu k veku žiakov
* Plnenie základných didaktických funkcií

## Hejného metóda

Hejného metóda je spôsob vyučovania matematiky, kde deti nie sú nútené mechanicky memorovať rozličné matematické postupy a vzorce. Práve naopak, pri riešení úloh sami prichádzajú na súvislosti a vzťahy medzi jednotlivými príkladmi. Pedagóg má pri tomto spôsobe vyučovania za úlohu správne deti nasmerovať, povzbudzovať ich k diskusii o riešeniach a hlavne mať dostatočnú trpezlivosť.

Za vznik Hejného metódy [2] vďačíme Vítovi Hejnému, ktorý skúmal príčinu, prečo sa žiaci len učia naspamäť vzorce a postupy, miesto toho, aby porozumeli princípu. V jeho práci ďalej pokračoval jeho syn, prof. Milan Hejný, ktorý začal spoločne s jeho spolupracovníkmi rozpracovávať poznatky svojho otca. Tieto poznatky boli vylepšené a publikované v roku 1987. V dnešnej dobe sa táto metóda nevyučuje len na mnohých školách na Slovensku a v Česku, ale aj v niektorých zahraničných krajinách.

Hejného metóda je založená na 12 základných princípoch [3], ktorými sú:

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

Každý človek, či už dospelý alebo dieťa, si v dôsledku potreby nevedome vytvára v hlave schémy. Schéma je súhrn navzájom prepojených znalostí, ktoré sa týkajú známeho prostredia. Tieto schémy sú ako v bežnom živote, tak aj v matematike hlavným nástrojom našich rozhodnutí. Napríklad ak by sa vás niekto spýtal, kde nájde vo vašom obľúbenom obchode ryžu, vedeli by ste mu odpovedať pomocou vašej schémy obchodu, ktorú máte v hlave. Hejného metóda tieto schémy napája na seba, posilňuje a vyvodzuje z nich konkrétne úsudky.

1. **Práca v prostrediach – Učíme sa opakovanou návštevou**

Hejného metóda je založená na práci s prostrediami. Každé prostredie je zamerané na iný matematický jav a väčšina prostredí je založená na reálnych skúsenostiach z bežného života detí. Silnou stránkou prostredí je motivácia detí k práci a učení. Dizajny jednotlivých prostredí sú pre nich lákavé a pútavé, čo navodzuje skôr pocit hrania ako učenia alebo pracovania. Vďaka tomu sa k prostrediam radi vracajú a objavujú nové úlohy a riešenia.

1. **Prelínanie tém – Matematické zákonitosti neizolujeme**

V rôznych prostrediach a úlohách sa deti učia jednotlivé pojmy, procesy, stratégie, vzájomné väzby a k ich dobrému zapamätaniu a porozumeniu dôjde pomocou poskladania akejsi mozaiky poznatkov z jednotlivých prostredí a činností.

Keby sme sa schému nášho obchodu učili s odstupom času, napríklad by sme sekciu pečiva prebrali v septembri, mäso v októbri a nápoje v novembri, tak v decembri by sme si museli opakovať všetko, čo sme už o pečive a mäse zabudli. No vďaka tomu, že obchod poznáme z vlastnej skúsenosti, sme si schopní celý obchod vybaviť aj s jeho oddeleniami.

1. **Rozvoj osobnosti – Podporujeme samostatné uvažovanie detí**

Škola má veľký vplyv nie len na vedomostný, ale aj na psychický a osobnostný rast dieťaťa. Je to prostredie, v ktorom trávia väčšinu času a odohráva sa tu množstvo kľúčových okamžikov. Hejného metóda je dôkazom toho, že dobré vyučovanie môže mať obrovský vplyv na osobnostný rast detí.

1. **Skutočná motivácia – Keď ,,neviem“ a ,,chcem vedieť“**

Motiváciu môžeme chápať ako potrebu poznávania nových vecí, ktorá pramení z vnútorného boja medzi ,,neviem“ a ,,chcem vedieť“. Je prirodzené, že každé dieťa je zvedavé a má potrebu spoznávať a porozumieť veciam, ktorým nechápe.

V tradičnej výuke je impulzom k učeniu pre žiaka snaha získať dobrú známku alebo skôr strach zo zlej známky. Hejného metóda ponúka deťom prirodzenú motiváciu.

1. **Reálne skúsenosti – Staviame na vlastných zážitkoch dieťaťa**

Každý si vie lepšie zapamätať veci, ktoré vyplývajú z našej reálnej skúsenosti. To isté platí aj pri učení matematiky. Výučba matematiky orientovaná na vybudovanie schém, ktoré vyplývajú z vlastných skúseností je lepšou alternatívou, ako memorovanie postupov, ktoré si dieťa nevie vysvetliť.

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

Keď deti prídu samé na správne riešenie, prináša im to radosť. Táto radosť sa potom stáva ich primárnou motiváciou pokračovať pri ďalších úlohách. Prostredia Hejného metódy sú navrhnuté tak, aby im umožnili toto objavovanie.

Žiak, ktorý je donútený k memorovaniu poznatkov a postupov a ktorému je odopreté vlastné objavovanie, nebude v budúcnosti schopný a ochotný dopracovať sa k poznatkom sám.

1. **Vlastný poznatok – Má väčšiu váhu ako ten prebratý**

Každý poznatok, na ktorý sme prišli vlastnou hlavou, je kvalitnejší ako poznatok, ktorý preberieme od niekoho iného. Každý jeden z nás máme iné postupy a skúsenosti, preto je dobré naše postupy zdieľať, rozprávať sa o nich a pomôcť tým ostatným, aby prišli na vlastný postup, ktorému budú rozumieť.

1. **Rola učiteľa – Sprievodca a moderátor diskusií**

Pri výučbe matematiky Hejného metódou má rola vyučujúceho iný význam ako pri bežnej výučbe. Učiteľ tu nemá byť videný ako len autorita, ktorá deťom vie a vysvetľuje učivo a deti si to zapisujú do zošitov. Aj keď učiteľ vie, čo sa deti idú učiť, nedáva im to najavo. Je to pre žiakov niekto, kto organizuje hodinu, podporuje diskusie a reaguje na situáciu medzi žiakmi.

1. **Práca s chybou – Predchádzame zbytočnému strachu detí**

Hovorí sa, že chybami sa učíme. Chyby sú prirodzenou súčasťou učenia sa nových vecí. Ak človek vie, že sa dopustil chyby a vie aj jej dôvod, je na dobrej ceste k tomu, aby sa jej nabudúce vyhol. Chyby nesmú žiakov odradiť alebo priviesť k strachu. Kým učiteľ berie chybu ako zlú, nežiadúcu vec, tak budú žiaci v nepokoji a strachu, čo im určite nepomôže k pochopeniu príčiny chyby. Pri Hejného metóde sú chyby vítané, lebo sú cestou ku skutočnému poznaniu.

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

Každý žiak má vlastné tempo a limity. Vyučujúci by mal preto zodpovedať za to, že šikovnejší aj slabší žiaci majú stále rovnakú motiváciu na riešenie úloh. Či už to znamená nastaviť slabším žiakom úlohy, ktoré zvládnu a vedia sa posunúť vlastným tempom na náročnejšie, alebo šikovnejším žiakom naopak zabezpečiť, aby sa nenudili pri úlohách, ktoré pre nich nie sú dostatočne náročné.

1. **Podpora spolupráce – Poznatky sa rodia vďaka diskusii**

Každé dieťa je iné a vyhovuje mu iný spôsob práce a riešenia. Sú deti, ktoré radi pracujú samostatne a niektorým sa lepšie pracuje v kolektíve alebo vo dvojici. Vzájomná diskusia je pre deti veľmi dôležitým aspektom pri nadobúdaní nových poznatkov. Diskusie motivujú deti k hľadaniu správneho riešenia.

## Prostredie Ježibaba Armida

Hejného metóda využíva viacero výukových prostredí, ktoré slúžia na to, aby deti radi riešili matematické úlohy a dobre porozumeli matematickým pojmom a vzťahom medzi nimi. Moja práca je zameraná na prostredie Ježibaba Armida [4], ktoré je určené pre žiakov štvrtého a piateho ročníka základnej školy. Prostredie Ježibaba Armida sa venuje práci so zápornými číslami a predstavuje žiakom diofantické rovnice. Umožňuje žiakom získať predstavu napríklad o tom, prečo keď vynásobíme dve záporné čísla dostávame kladné číslo.

V úlohách pracujú deti s horúcimi a studenými kameňmi, ktoré reprezentujú kladné (horúci) a záporné (studený) čísla. Ježibaba Armida varí elixír, ktorý má určitú teplotu. Ak do kotlíka s elixírom prihodíme studený kameň, teplota elixíru sa zníži o jeden stupeň. Ak tam prihodíme horúci, tak teplota stúpne o jeden stupeň. Pomocou kameňov tak vieme ovplyvňovať teplotu elixíru alebo zistiť, akú má elixír teplotu ak doň Armida hodila určité kamene.



Obrázok 1: Príklad úlohy z prostredia Ježibaba Armida



Obrázok 2: Príklad úlohy z prostredia Ježibaba Armida

Typické úlohy v tomto prostredí vyzerajú napríklad nasledovne:

*Elixír mal na začiatku teplotu 14 stupňov. Ježibaba Armida prihodila horúci kameň s hodnotou 4 a studený kameň s hodnotou 2. Zisti, akú teplotu mal elixír na konci.*

*Elixír má teplotu 32 stupňov. Máme k dispozícii horúci kameň s hodnotou 3 a horúci kameň s hodnotou 4. Ježibaba Armida potrebuje, aby mal elixír teplotu 44 stupňov, pomôž jej aby dosiahla požadovanú teplotu.*

*Elixír má teplotu 8 stupňov. Máme k dispozícii horúci kameň s hodnotou 2 a studený kameň s hodnotou 4. Ježibaba Armida potrebuje, aby mal elixír teplotu 26 stupňov, pomôž jej aby dosiahla požadovanú teplotu.*

Určite tušíte, že ide o diofantické rovnice, ktoré si priblížime v nasledujúcej kapitole.

## Diofantické rovnice

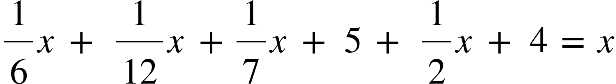
V mojej práci som sa venovala diofantickým rovniciam, preto priblížim ich význam.

### 1.4.1 Diofantos a vznik diofantických rovníc

Diofantické rovnice [8] nesú meno gréckeho aritmetika Diofanta z Alexandrie. O jeho živote sa nezachovalo veľa informácií. Vieme, že žil približne v treťom storočí a veľkú časť svojho života prežil v Alexandrii. Taktiež vieme dĺžku jeho života, keďže sa na jeho náhrobnom kameni nachádza matematická hádanka:

*Diofantova mladosť trvala 1/6 jeho života. Fúzy mu narástli o ďalšiu 1/12 jeho života. O nasledujúcu 1/7 života sa Diofantos oženil. Po piatich rokoch sa mu narodil syn. Syn žil presne ½ dĺžky života svojho otca. Diofantos zomrel 4 roky po smrti svojho syna.*[[1]](#footnote-1)

a pomocou nej sa dopracujeme k rovnici:



Po vyriešení rovnice sa dostaneme k výsledku, že Diofantos žil 84 rokov.

Za jeho život napísal mnoho diel, z ktorých sa do dnes zachovala len časť z nich. Jeho najvýznamnejšie dielo s názvom *Aritmetika*, kde sa venoval riešeniu lineárnych a kvadratických rovníc, mu doprialo označenie ,,otec algebry´´. *Aritmetika* sa skladá z 13 kníh, z ktorých sa zachovalo len šesť. Tieto knihy obsahovali zbierku rovníc, dnes známych ako diofantické rovnice, čiže rovnice s celočíselnými koeficientami, ktoré majú riešenia z oboru celých čísiel. Diofantos napísal aj spisy Diofantické rovnice a Diofantické čísla, v ktorých predstavil rovnice o dvoch neznámych, v tvare:

a x space plus space b y space equals space c, (1)

kde *a*, *b*, *c* sú celočíselné konštanty a *x*, *y* sú neznáme z oboru celých čísiel.

Spolu s Euklidovými zväzkami s názvom *Základy* [9] bola *Aritmetika* základom matematiky až do 17. storočia.

David Hilbert, významný matematik, ktorý žil v 19-20. storočí, predložil v roku 1900 na matematickej konferencii v Paríži zoznam 23 nevyriešiteľných problémov dôležitých pre vtedajšiu matematiku. Je veľmi podstatné ho spomenúť v mojej práci práve preto, že jeho desiaty problém sa týka diofantických rovníc, konkrétne problému rozhodnuteľnosti (resp. existencie algoritmu), či má ľubovoľná diofantická rovnica riešenie. [10]

Je nemožné vytvoriť univerzálnu metódu na výpočet diofantických rovníc. Toto dokázal v roku 1970 ruský matematik Jurij Matijasevič vo svojej diplomovej práci pomocou Fibonacciho postupnosti[[2]](#footnote-2).

Lineárna diofantická rovnica má tvar:

a subscript 1 x subscript 1 space plus space a subscript 2 x subscript 2 space space plus space... space plus space a subscript n x subscript n space space equals space b, (2)

kde *x1, x2, ..., xn* sú celočíselné neznáme, *a1, a2, ..., an* a *b* sú celočíselné koeficienty rovnice. V úlohách, ktoré generujem v mojej aplikácii sa stretávame s lineárnymi diofantickými rovnicami o dvoch a troch neznámych, teda s rovnicami v tvaroch:

a x space plus space b y space equals space c (3)

a x space plus space b y space plus space c z space equals space d, (4)

kde *x, y, z* sú celočíselné neznáme, *a, b, c* a *d* sú celočíselné koeficienty rovnice. Riešením týchto rovníc (3) a (4) sú n-tice celých čísiel (x, y) a (x, y, z), ktoré vyhovujú rovniciam (3) a (4). Ako som spomínala, neexistuje univerzálna metóda na riešenie všeobecných diofantických rovníc. Avšak existuje niekoľko spôsobov, pomocou ktorých vieme vyriešiť lineárne rovnice tvaru (3) a (4).

### 1.4.2 Riešenie pomocou rozšíreného Euklidovho algoritmu

Najdôležitejšiu úlohu pri tomto riešení lineárnych diofantických rovníc hrá deliteľnosť čísel, konkrétne nájdenie najväčšieho spoločného deliteľa. Najväčší spoločný deliteľ je najväčšie kladné číslo, ktoré delí obe čísla (a, b). Na výpočet najväčšieho spoločného deliteľa použijeme Euklidov algoritmus. Vďaka tomuto poznatku vieme zistiť, či má lineárna diofantická rovnica aspoň jedno riešenie alebo nemá žiadne. Majme rovnicu tvaru (3), ak najväčší spoločný deliteľ (a, b) delí c, tak rovnica má aspoň jedno riešenie. V opačnom prípade rovnica riešenie nemá [11]. Ukážme si to na príkladoch :

Príklad:

4 x space minus 2 y space equals space 12

najväčší spoločný deliteľ čísiel 4, 2 = 2 a 2 delí 12 a vieme ľahko overiť, že rovnica má aspoň jedno riešenie, napríklad dosadíme za x = 4 a y = 2:

4 times bold 4 space minus space 2 times bold 2 equals space 12
16 space minus space 4 space equals space 12
12 space equals space 12

Príklad:

4 x space minus space 2 y space equals space 7

najväčší spoločný deliteľ čísiel 4,2 = 2 a 2 nedelí 7, teda rovnica nemá ani jedno riešenie na množine celých čísel. Môžeme si to skontrolovať aj bežnou logikou. Ľubovoľnou kombináciou dvoch párnych čísiel vždy dostaneme párne číslo, teda neexistuje kombinácia, kde z čísiel 4,2 dostaneme nepárne číslo 7.

Pri rovnici tvaru (4) musíme opakovane použiť Euklidov algoritmus, teda postup, kde najprv určíme najväčšieho spoločného deliteľa a, b (označíme ako NSD(a, b)) a potom určíme najväčšieho spoločného deliteľa NSD(a, b) a c, teda nasledovne: NSD(a, b, c).

Teraz vieme zistiť, či má naša rovnica aspoň jedno riešenie, alebo či nemá žiadne riešenie. Ďalej by sme potrebovali zistiť, ako vieme tieto riešenia efektívne zisťovať, prípadne ovplyvniť to, či chceme generovať rovnicu s nekonečným alebo nulovým počtom riešení. S týmto mi pomohol rozšírený Euklidov algoritmus[[3]](#footnote-3) (extended Euclidean algorithm). Je to postup, ktorý pre zadanú dvojicu prirodzených čísiel a, b vypočíta ich najväčší spoločný deliteľ NSD(a, b) a celé čísla x, y také, že platí:

a x space plus space b y space equals space N S D open parentheses a comma space b close parentheses, (5)

kde čísla *x, y* nazývame tiež aj Bézoutove koeficienty rovnosti [4]. Bézoutove koeficienty *x, y* pre *a, b* nie sú unikátne, je to len jedno možné riešenie rovnice. Toto riešenie vieme získať pomocou rozšíreného Euklidovho algoritmu, a to spätným dosadzovaním do Euklidovho algoritmu nasledovne:

Príklad: 7 x space plus space 4 y space equals space 31 (6)

Najprv si rozpíšeme Euklidov algoritmus:

7 space equals space 4 space times space bold 1 space plus space 3
4 space equals space 3 space times space bold 1 space plus space 1
3 space equals space 1 space times space bold 3 space plus space 0

Posledný nenulový zvyšok je 1, čiže vieme, že najväčší spoloční deliteľ čísiel 7 a 4 je 1 (NSD(7,4) = 1). Pre dobrú viditeľnosť funkcie algoritmu upravíme rovnice nasledovne:

3 space equals space 7 space minus space 4 space times space 1
1 space equals space 4 space minus space 3 space times space 1

Následne získame Bézoutove koeficienty pomocou spätného dosádzania:

3 space equals space 7 space minus space open parentheses 4 space times space 1 close parentheses
1 space equals space 4 space minus space open parentheses 7 space minus space 4 space times space 1 close parentheses space times space 1
1 space equals space 4 space minus 7 space minus space open parentheses 4 space times space 1 close parentheses
1 space equals space open parentheses 4 space times space 2 close parentheses space minus space open parentheses 7 space times space 1 close parentheses

Vidíme, že sme dostali rovnicu tvaru (5), kde [x, y] = [1, 2] a NSD(a, b) = 1. Pomocou rozšíreného Euklidovho algoritmu sme zistili najväčšieho spoločného deliteľa čísiel 7 a 4, ďalej sme zistili Bézoutove koeficienty x, y. Ak chceme dostať koeficienty pre našu zadanú rovnicu (6), vynásobíme obe strany rovnice nasledovne:

1 space equals space open parentheses 4 space times space 2 close parentheses space minus space open parentheses 7 space times space 1 close parentheses
7 space times space open parentheses negative 1 close parentheses space plus space open parentheses 4 space times space 2 close parentheses space equals space 1 space space space space divided by 31
7 times space open parentheses bold minus bold 31 close parentheses space plus space 4 space times space bold 62 space equals space 31

Nájdené Bézoutove koeficienty *x0* (-31) a *y0* (62) sú veľmi dôležitou súčasťou nájdenia všetkých možných riešení. Ak existuje aspoň jedna usporiadaná dvojica koeficientov *x0*a *y0*, potom existuje nekonečne veľa ďalších koeficientov [8] nasledovne:

x space equals space x subscript 0 space plus space fraction numerator b over denominator N S D open parentheses a comma b close parentheses end fraction times t
y space equals space y subscript 0 space minus space fraction numerator a over denominator N S D open parentheses a comma b close parentheses end fraction times t t space element of space Z (7)

Môžeme to opäť vyskúšať na našom príklade (6). Už vieme, že *x0* = -31 a *y0* = 62, takisto vieme, že a = 7, b = 4 a NSD(a, b) = 1 dosadíme do vzorcov (7):

x space equals space minus 31 space plus space 4 over 1 times t
y space equals space 62 space minus space 7 over 1 times t


p r e space t space equals space 0 colon space space space x space equals space minus 31 space plus space 4 times 0 space equals space minus 31 space space space space space y space equals space 62 space minus space 7 times 0 space equals space 62 space space space space space 7 space times space open parentheses bold minus bold 31 close parentheses space plus space 4 space times space bold 62 space equals space 31
p r e space t space equals space 1 colon space space space x space equals space minus 31 space plus space 4 times 1 space equals space minus 27 space space space space space y space equals space 62 space minus space 7 times 1 space equals space 55 space space space space space 7 times space open parentheses bold minus bold 27 close parentheses space plus space 4 space times space bold 55 space equals space 31
p r e space t space equals space 2 colon space space space x space equals space minus 31 space plus space 4 times 2 space equals space minus 23 space space space space space y space equals space 62 space minus space 7 times 2 space equals space 48 space space space space space 7 times space open parentheses bold minus bold 23 close parentheses space plus space 4 space times space bold 48 space equals space 31

Vidíme, že pomocou Euklidovho algoritmu vieme pekne zistiť, či má lineárna diofantická rovnica riešenie. Pomocou rozšíreného Euklidovho algoritmu a Bézoutových koeficientov vždy dokážeme nájsť aspoň jedno jej riešenie, vďaka ktorému po dosadení do vzorca Bézoutovej rovnosti vieme nájsť aj ďalšie riešenia. Presne tieto poznatky som použila pri generovaní úloh v prvých dvoch úrovniach, kde sa stretávame s rovnicami o dvoch neznámych. Spôsobov na riešenie lineárnych diofantických rovníc však existuje viac ako len rozšíreným Euklidovým algoritmom. Stručne si prejdeme niektoré z najznámejších spôsobov.

### 1.4.3 Riešenie pomocou kongruencie

Pojem kongruencie je veľmi dôležitý v teórii čísiel. Tento pojem zaviedol nemecký matematik Carl Friedrich Gauss (1777-1855) vo svojej knihe *Disquisitiones Arithmeticae* [13].

Definícia:

Majme ľubovoľné dve celé čísla a, b a číslo n, také, že n je z prirodzených čísiel a n ≥ 2. Potom hovoríme, že a, b sú kongruentné podľa modulo n práve vtedy, ak a-b je deliteľné číslom n. Dve čísla kongruentné podľa modulo n dávajú po delení týmto modulom rovnaký zvyšok [10]. Zapisujeme nasledovne:

a space identical to space b space open parentheses m o d space n close parentheses (8)

Túto metódu si ukážeme na príklade (6). Keďže musia byť *x, y* celé čísla, musí platiť:

x space equals space fraction numerator 4 y space minus space 31 over denominator 7 end fraction
y space equals space fraction numerator 7 x space minus space 31 over denominator 4 end fraction (9)

Lineárnu diofantickú rovnicu o dvoch neznámych môžeme transformovať na kongruenciu o jednej neznámej. Potom už záleží na nás, podľa ktorej neznámej budeme kongruenciu počítať, výsledok bude rovnaký. Zvolená neznáma bude pre nás neznámou a druhá nám poslúži ako modul, podľa ktorého budeme kongruenciu počítať. V našom príklade budeme počítať neznámu x pomocou y nasledovne:

7 x space identical to space 31 open parentheses m o d space 4 close parentheses (10)

Vznikne nám aj pomocná rovnica:

4 x space identical to space 0 open parentheses m o d space 4 close parentheses (11)

Ďalšie kroky spočívajú v odpočítavaní kongruencií (10) a (11) s dodatočnými úpravami tak, aby sme dostali na ľavej strane kongruencie x:

7 x space identical to space 31 open parentheses m o d space 4 close parentheses
4 x space identical to space 0 open parentheses m o d space 4 close parentheses

Odčítaním prvej kongruencie od druhej získame tvar:

3 x space identical to space 31 open parentheses m o d space 4 close parentheses (12)

Opäť odčítame kongruenciu (12) od kongruencie (11) a získame tvar:

negative x space identical to space 31 open parentheses m o d space 4 close parentheses (13)

Výslednú kongruenciu (13) vynásobíme -1 a dostaneme tvar:

x space identical to space minus 31 open parentheses m o d space 4 close parentheses (14)

Danú kongruenciu vieme previezť na rovnicový tvar:

x space equals space minus 31 space plus space 4 t, (15)

kde t je z množiny celých čísiel.

Pre získanie neznámej y dosadíme (15) do rovnice (9):

y space equals space fraction numerator 7 open parentheses negative 31 space plus space 4 t close parentheses space minus 31 over denominator 4 end fraction space equals space fraction numerator negative 217 space plus space 28 t space minus space 31 over denominator 4 end fraction
y space equals space fraction numerator negative 248 space plus space 28 t over denominator 4 end fraction space equals space minus 62 space plus space 7 t
y space equals space minus 62 space plus space 7 t


kde t je z množiny celých čísiel.

Tento spôsob riešenia pomocou kongruencie [10] je len jedným možným spôsobom. Iný spôsob riešenia vieme nájsť napríklad v práci [8].

### 1.4.4 Riešenie pomocou redukčnej metódy

Princíp redukčnej metódy (nazývaná aj ako vyjadrenie členov s najmenším koeficientom) spočíva v tom, že sa snažíme pôvodnú rovnicu čo najviac upraviť za predpokladu, že vieme, že výsledkom rovnice majú byť celé čísla. Tento spôsob riešenia je bližšie opísaný v práci [11].

### 1.4.5 Diofantické rovnice o troch neznámych

Pri práci s rovnicami o troch neznámych sa bohužiaľ metóda rozšíreného Euklidovho algoritmu použiť nedá, keďže nie vždy je schopná presne určiť všetky riešenia [11]. Preto som pri tomto generátore použila len prácu s Euklidovým algoritmom a následne pomocou techniky backtrackingu[[4]](#footnote-4) našla všetky potrebné riešenia.

Dajú sa však použiť metódy kongruencie a redukčná metóda. Riešenie diofantických rovníc o troch neznámych pomocou týchto metód nájdeme v práci [8].

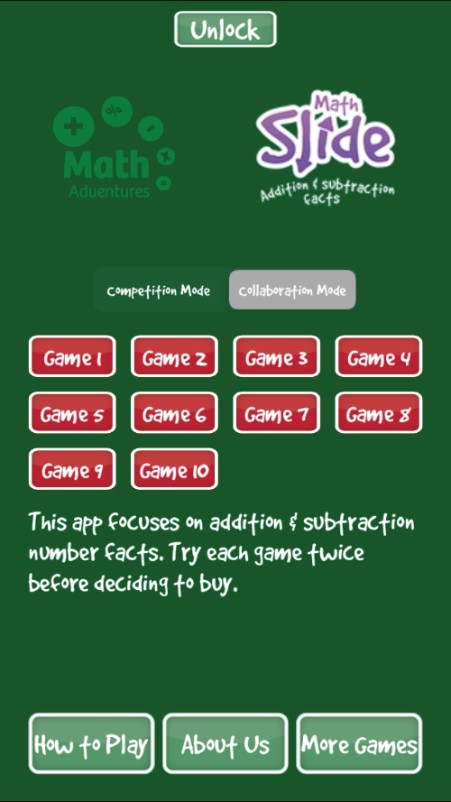
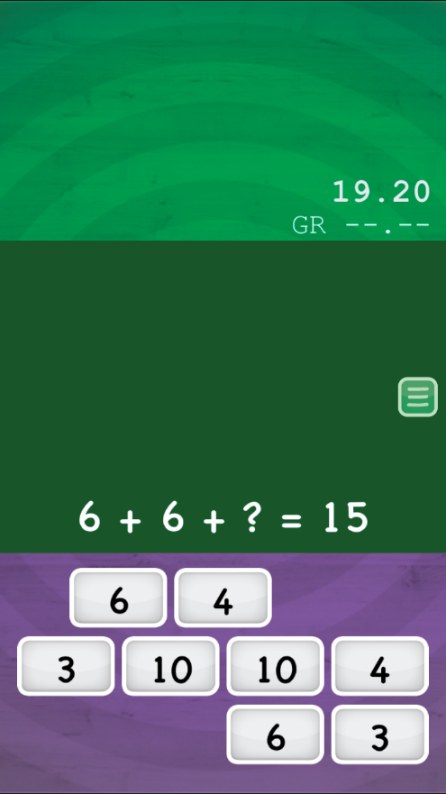
## Podobné práce

V tejto časti predstavím príklady existujúcich implementácií matematických aplikácií založených na princípe učenia hrou alebo na Hejného metóde. Prejdem dve mobilné aplikácie: **Math Slide [5]** a **Matemág [6]**, obe dostupné pre operačné systémy Android aj IOS a taktiež spomeniem jednu minuloročnú bakalársku prácu, zameranú na prostredie **Krokovanie [7]**, ktoré je príbuzné prostrediu Ježibaba Armida.

### Math Slide

Math slide je séria aplikácií založených na princípe učenia hrou. V tejto sérii matematických hier nájdeme aplikácie sústredené na sčitovanie a odčitovanie, delenie a násobenie a na základné matematické fakty. Bližšie predstavím len aplikáciu sústredenú na sčitovanie a odčitovanie.

Po spustení aplikácie a prejdení cez uvítaciu obrazovku sa dostaneme k hlavnému menu. Menu disponuje desiatimi tlačidlami na zvolenie úrovne a troma tlačidlami, ktoré poskytujú viac dodatočných informácií napríklad o tom, ako aplikácia funguje.

Obrázok 3: Math Slide menu Obrázok 4: Math Slide hra

Každá úroveň je navrhnutá pre jedného alebo dvoch hráčov. Ak hrajú dvaja hráči, aplikácia je navrhnutá ako súťaživá hra medzi oboma hráčmi. Všetky úlohy sú zamerané na hráčovu schopnosť rýchleho počítania. Aplikácia sa ľahko ovláda, stačí len čísla potiahnuť do stredu obrazovky. S každou úrovňou graduje zložitosť úloh a každá úroveň má časovač, ktorý kontroluje náš časový rekord v danej úrovni. Nevýhodou je, že aplikácie je dostupná len v anglickom jazyku.

### Matemág

Matemág je príbehová edukačná aplikácia, kde sledujeme príbeh súrodencov Terezky a Kubka, ktorí sa snažia dostať k čarodejníkovi. Na ich ceste za čarodejníkom musia prekonávať rôzne matematické prekážky tvorené z rozličných prostredí Hejného metódy.

Ak zapíname aplikáciu po prvýkrát, ukáže sa nám obrazovka, kde máme zadať naše meno, jazyk a ročník základnej školy. Po vyplnení údajov prechádzame na hlavné menu, ktoré disponuje pútavou grafikou a animáciami. V menu vidíme niekoľko uzamknutých úrovní v podobe ostrovov a tlačidlo nastavení, kde si vieme napríklad vypnúť zvuk alebo zmeniť jazyk. Aplikácia dovoľuje nastavenie mnohých jazykov, medzi ktorými sú aj slovenský a český jazyk.



Obrázok 5: Matemág, tretia úroveň

Počas hrania úrovní môžeme sledovať aj príbeh, ktorý sa nám postupne odomyká. Príbeh je robený formou komiksu a animácií. Matemág je veľmi dobre spracovaná edukačná aplikácia, ktorá okrem svojej pútavej grafiky a animácií, obsahuje aj zaujímavý príbeh a intuitívne ovládanie.

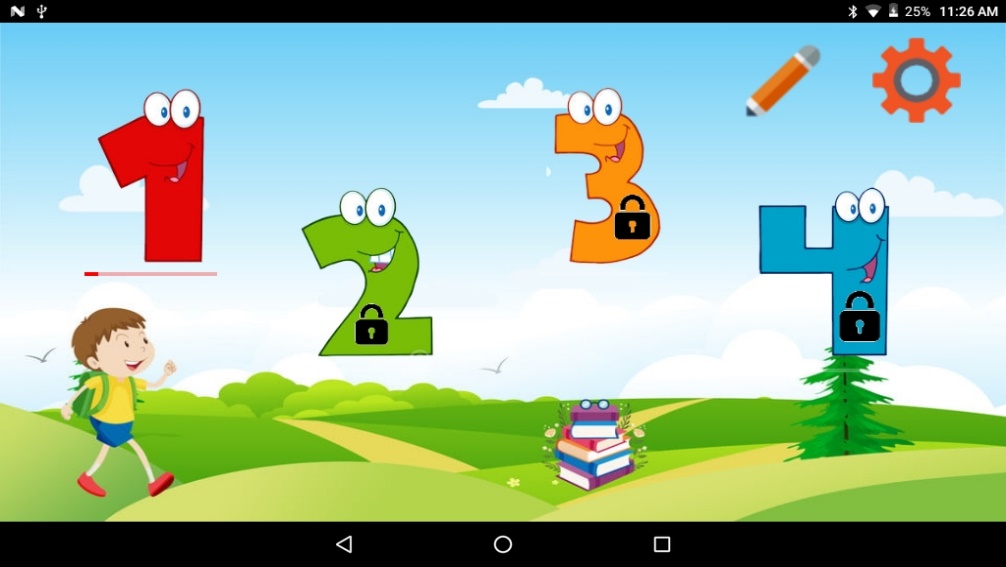


Obrázok 6: Matemág, komiks

### Krokovanie

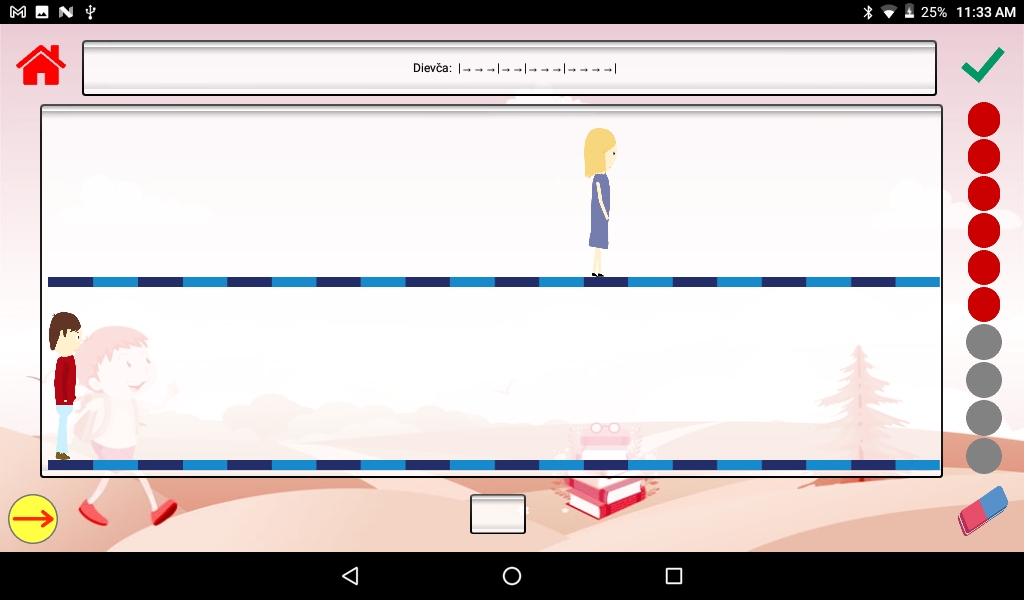
Krokovanie je jedným z prostredí Hejného metódy, kde ide o prepojenie pohybu a počítania. Základom je sčítanie a odčítanie krokov, pri ktorých sa používa krokovací pás. Bakalársku prácu, ktorá sa zaoberá prostredím krokovanie, vytvorila v roku 2022 Claudia Barillová.

Aplikácia má pestré hlavné menu, kde vidíme štyri tlačidlá možných úrovní, prvá úroveň je odomknutá a ostatné tri sú uzamknuté. Z menu sa vieme dostať aj na obnovenie úrovní a na výber editora úloh, kde je pre každú úroveň editor odomknutý až po dokončení danej úrovne.



Obrázok 7: Krokovanie, hlavné menu

Úrovne aj úlohy sú v aplikácii dobre rozvrhnuté, začíname na jednoduchých krokových úlohách. V týchto úlohách hráme za chlapca, pomocou ktorého krokujeme. Krokovanie je spravené pomocou animácií a naše kroky sa zobrazujú na spodnej časti obrazovky v podobe šípok, podľa toho, ktorým smerom sa hýbeme. Krokovací pás je prehľadne spravený, pomocou dvoch odlišných striedavých farieb. V neskorších úrovniach sledujeme aj krokovanie dievčatka. Neskôr prídu aj rovnice zobrazené pomocou šípok. Pri niektorých úlohách bolo dopĺňanie šípok do rovnice neintuitívne.



Obrázok 8: Krokovanie, prvá úroveň

Hra je zaujímavá vďaka animáciám, ktoré dodávajú efekt krokovania. Aplikácia sa až na pár nedostatkov s dopĺňaním neznámych dobre ovláda. Na obrazovke nie sú žiadne zbytočné predmety alebo tlačidlá, hra je intuitívna a prehľadná.

# Návrh

V tejto kapitole sa budem venovať návrhu používateľského rozhrania, teda návrhov jednotlivých obrazoviek úrovní v mojej aplikácii. Ďalej prejdem návrh úrovní, úloh a editorov úloh.

## Používateľské rozhranie a logika úrovní

Návrh používateľského rozhrania je veľmi dôležitým aspektom pri vytváraní edukačného softvéru. Keďže sa jedná o prácu s deťmi, pri ktorých je veľmi podstatné, aby ich zaujal dizajn aplikácie, treba mu venovať dostatok času a kreativity. Používateľské rozhranie by malo pútať, no nie vyrušovať. Teda aby sa vedeli deti sústrediť na podstatu riešenia úlohy a nie na samotné komponenty v aplikácii.

Dizajn aplikácie je navrhnutý na šírku (landscape) a rozlíšenie sa prispôsobí zariadeniu, teda veľkosti všetkých prvkov sa prispôsobia veľkosti obrazovky. Pre ideálny zážitok je odporúčaným zariadením tablet, resp. zariadenie s väčšou obrazovkou. Každá obrazovka, či už menu, playground alebo úroveň, disponuje rovnakým pozadím s animáciou ohňa v krbe a taktiež ježibabou Armidou a animáciou miešania v kotlíku. Pri každej obrazovke vidíme v pravom hornom rohu kruhové tlačidlo s písmenom ,,**i**“, po ktorého stlačení sa spustí vysvetlenie jednotlivej úrovne v podobe hlasu Armidy.

Aplikácia je rozdelená do štyroch úrovní, ktorých náročnosť postupne graduje. V každej úrovni je k dispozícii desať úloh, ktorých náročnosť taktiež postupne stúpa. Ak hráč zapína aplikáciu po prvýkrát, je mu dostupná len prvá úroveň. Ostatné úrovne sa postupne odomykajú po vyriešení aspoň polovice úloh v predchádzajúcej úrovni. Ak hráč vyriešil polovicu úloh v danej úrovni, spustí sa animácia a audio s oznamom odomknutia ďalšej úrovne. Pri návrhu úrovní som sa inšpirovala najmä úlohami z učebnice pre základné školy [14], ktorá obsahuje úlohy zamerané na Hejného metódu výučby matematiky. Keďže úlohy tohto typu sú prezentované žiakom, teda sú im známe, bude sa im ľahšie s aplikáciou pracovať.

### Menu

Po prvom spustení aplikácie sa nám zobrazí menu (Obr. 9), kde vidíme pozadie s obrázkom krbu a animáciou ohňa, ježibabu a nadpis aplikácie ,,Ježibaba Armida“. V strede obrazovky sa nachádzajú tlačidlá jednotlivých úrovní. Tlačidlá všetkých úrovní majú podobu fľašiek s elixírom a číslom danej úrovne. Keďže sme aplikáciu zapli prvýkrát, vidíme len tlačidlo prvej úrovne a ostatné úrovne vidíme zamknuté. Za tlačidlami úrovní vidíme tlačidlo s nápisom ,,Playground“.



Obrázok 9: Hlavné menu s uzamknutými úrovniami

Ďalej vidíme tri kruhové tlačidlá, jeden v pravom hornom rohu, ktorý sme už spomínali, a dva v pravom dolnom rohu. Prvé z tlačidiel slúži na nastavenie hry, konkrétnejšie na obnovenie jednotlivej úrovne. To sa dá jedine pre odomknutú úroveň, ak je zamknutá, vidíme tlačidlo na obnovenie ako priesvitno-sivé. Druhé z tlačidiel slúži na vypnutie aplikácie.



Obrázok 10: Hlavné menu aplikácie po otvorení všetkých úrovní

### Úrovne 1 a 2

Prvá úroveň vyžaduje iba poznatok základného sčitovania záporných a kladných čísiel. Na teplomere je zobrazená teplota elixíru predtým, ako doň prihodila Armida kamene, ktoré sú zobrazené na paneli. Hráč musí zadať správnu výslednú teplotu elixíru, teda pričítať hodnoty kameňov k teplote na teplomere. V tejto úrovni postupne graduje hodnota kameňov a aj počet kameňov na paneli.

Druhá úroveň má rovnaký princíp ako prvá, až na to, že už používame aj odčitovanie kladných aj záporných čísiel a používame aj zátvorky. Pri tejto úrovni je hlavným cieľom, aby deti pochopili koncept práce so zápornými číslami. Teda napríklad, keď odčitujeme záporné číslo a dostaneme kladné, alebo keď máme mínus pred zátvorkou a menia sa znamienka všetkých kameňov v zátvorke. Na paneli teda nájdeme už aj znamienka a zátvorky a opäť hľadáme výslednú teplotu elixíru. V tejto úrovni taktiež postupne graduje hodnota kameňov aj ich počet. V prvých úlohách vidíme len znamienko plus pred zápornými kameňmi, no postupne pred nimi pribúda aj znamienko mínus. Zátvorky sa taktiež objavujú až postupne, najprv sa pred nimi objavuje len znamienko plus a neskôr aj mínus. Aj počet kameňov v zátvorke postupne graduje.

Po spustení prvej alebo druhej úrovne uvidíme okrem už predtým spomenutých komponentov, v ľavom hornom rohu kruhové tlačidlo, slúžiace na návrat do menu.



Obrázok 11: Obrazovka prvej úrovne

Ďalej sa na ľavej časti obrazovky nachádza teplomer, ktorý patrí medzi hlavné komponenty v aplikácii a slúži na zobrazenie teploty elixíru v kotlíku. Teplotu vieme z teplomera vyčítať, no je aj zobrazená na jeho spodnej časti. V hornej časti vidíme tenký zelený obdĺžnik, slúžiaci ako ukazovateľ progresu (ďalej progress bar) hráča v danej úrovni. Progress bar sa začne plniť podľa počtu vyriešených úloh. V pravej časti obrazovky sa nachádza panel s kameňmi. Kamene reprezentujú buď záporné alebo kladné čísla. Ak vidíme červený, teda horúci kameň, znamená to, že sa jedná o kladné číslo. Ak naopak vidíme modrý, teda studený kameň, jedná sa o záporné číslo. Pri prvej úrovni vidíme na paneli len samotné kamene v poradí, ako ich Armida prihodila do kotlíka.



Obrázok 12: Obrazovka druhej úrovne

Pri druhej úrovni už vidíme na paneli aj znamienka a dokonca aj zátvorky. Teda niektoré kamene aj vybrala alebo prihodila v inom poradí. V prvej a druhej úrovni hráč nevie hýbať s kameňmi, tie len ukazujú, v akom poradí ich Armida hodila do kotlíka. Komponent na zadanie riešenia úlohy vidíme pod panelom s kameňmi. Výslednú teplotu vieme zadať pomocou posúvača (ďalej už len slider) a naša zadaná teplota sa ukazuje v kruhu nad ním. Posledný komponent sa nachádza v pravom dolnom rohu a je to tlačidlo, ktoré slúži na kontrolu úlohy resp. na prejdenie na ďalšiu úlohu, ak sme správne vyriešili predchádzajúcu. Farba komponentu na zadanie nášho riešenia je počas zadávania vždy oranžová. Ak sme zle zadali výsledok, tak sa jeho farba zmení na červenú a po dobe dvoch sekúnd sa naspäť zmení na oranžovo. Taktiež sa objaví obrázok bubliny s textom ,,Skús to ešte raz!“ a zaznie aj audio ježibaby Armidy. Ak sme zadali dobrý výsledok, tak sa farba zmení na zelenú, spustí sa animácia konfiet z kotlíka a opäť zaznie audio aj sa objaví bublina, tentoraz s textom ,,Dobrá práca!“ a prejde sa na ďalšiu úlohu.

### Úrovne 3 a 4

Tretia úroveň predstavuje nový princíp úloh. Vieme ju rozdeliť na dva typy úloh. Pri oboch typoch úloh sa na teplomere zobrazuje aktuálna teplota elixíru a bielou šípkou je zobrazená teplota, ktorú chceme dosiahnuť.

Pri prvom type úloh, ktoré sa nachádzajú v prvej polovici úrovne, je úlohou hráča nakombinovať dostupné kamene tak, aby sme dosiahli požadovanú teplotu. V tejto úrovni však nestačí nájsť jedno riešenie, treba nájsť všetky správne riešenia. Počet správnych riešení graduje každou druhou úlohou. Kamene treba presunúť do kotlíka pomocou funkcie ,,drag and drop“ a automaticky sa zmení hodnota na teplomere podľa pridaného kameňa. Ak sa teplota na teplomere rovná hodnote, ktorá je znázornená bielou šípkou, môžeme skontrolovať riešenie, a ak sme ho ešte nezadali, pridá sa nám medzi zadané riešenia. Ak sme riešenie už zadali, vypíše sa bublina a riešenie sa nepridá. Ak nájdeme všetky správne riešenia, môžeme prejsť na ďalšiu úlohu. Počet kameňov v rámci tohto typu úloh je najskôr dva, no neskôr tri.

Pri druhom type úloh, ktoré sa nachádzajú v druhej polovici úrovne, je opäť úlohou hráča nakombinovať dostupné kamene, no pri týchto úlohách máme konečný počet dostupných kameňov, ktoré musíme správne nakombinovať. V tomto type úloh sa nachádzajú aj záporné kamene. Pre prípad, že by sa hráč pomýlil a zle nakombinoval kamene, je dostupné tlačidlo na obnovenie úlohy. Naše zadané riešenia si vieme počas úlohy pozrieť a skontrolovať, stlačením tlačidla vedľa počtu riešení.

Štvrtá úroveň je vo väčšine zhodná s prvým typom úloh v tretej úrovni. Podstatným rozdielom je to, že sa tu nachádzajú úlohy, ktoré majú aj nekonečný počet riešení alebo nemajú žiadne riešenie. Pre zadanie riešenia týchto úloh máme dve nové tlačidlá, jedno na zadanie riešenia úloh s nekonečným počtom riešení a ďalšie na zadanie riešenia úloh, ktoré nemajú riešenie. Pri každej úlohe vidíme správny počet riešení, aj keď má úloha nekonečne veľa riešení alebo nemá žiadne. Ide o to, aby sme deťom neprezradili, že môže ísť o taký typ úlohy. Preto sa aj tlačidlá na zadanie týchto úloh objavia až dvadsať sekúnd po začiatku úlohy.

Pri obrazovkách tretej (Obr.13) a štvrtej (Obr.16) úrovne nájdeme už predtým spomenuté komponenty akými sú kruhové tlačidlá v horných rohoch obrazovky, progress bar a ježibaba Armida s rovnakým pozadím a animáciami. Teplomer je v týchto úrovniach upravený, resp. je k nemu pridaný komponent, ktorý ukazuje výslednú teplotu elixíru, na ktorú sa chceme dostať aby sme úlohu úspešne vyriešili. Tento komponent vidíme v podobe šípky a plochy na zobrazenie číselnej hodnoty teploty, na ktorú šípka ukazuje.

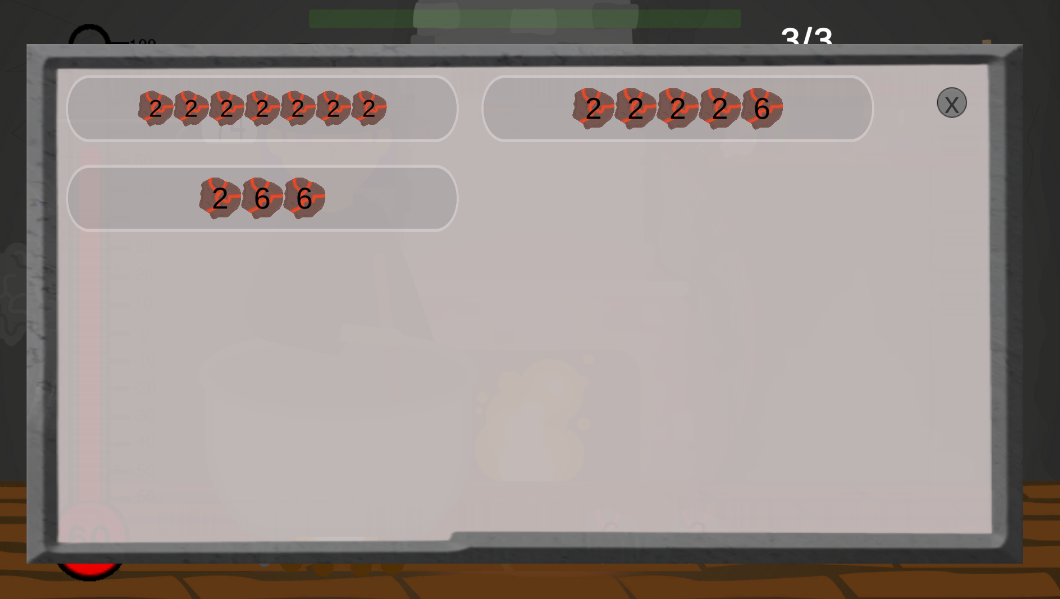
Ďalej vidíme napravo v hornej časti obrazovky text, ktorý je zapísaný v tvare *x/y*, kde *x* je zadaný počet správnych riešení a *y* je celkový počet správnych riešení. Vedľa textu sa nachádza kruhové tlačidlo s ikonou fľaštičky elixíru, po ktorého kliknutí sa nám zobrazí menšia obrazovka, kde sa nachádzajú naše správne zadané riešenia (Obr. 14).



Obrázok 13: Obrazovka tretej úrovne

Kamene už nevidíme na paneli, ale na spodnej časti obrazovky. Pri prvej polovici úloh tretej úrovne, môžu byť na ploche naraz dva alebo tri kamene. Pri druhej polovici úloh (Obr. 15) ich môže byť viacej a kamene sa nachádzajú na pravej časti obrazovky. Na obrazovke sa pri týchto úlohách objaví aj kruhové tlačidlo, ktoré obnovuje konfiguráciu celej úlohy pri prípade, že sme sa pomýlili.

V pravom spodnom rohu obrazovky sa nachádza štvorcové tlačidlo s ikonou fajky, ktoré musíme stlačiť, ak chceme skontrolovať práve zadané riešenie v kotlíku. Ak sa teplota elixíru nerovná požadovanej teplote, objaví sa obrázok bubliny s textom ,,Skontroluj si zadanú teplotu“. Ak by sme nemali žiadne kamene v kotlíku a chceme skontrolovať teplotu, objaví sa bublina s textom ,,Musíš najprv naplniť kotlík“. Naopak, ak sme dosiahli požadovanú teplotu, bublina bude obsahovať text ,,Len tak ďalej“ a priráta sa nám počítadlo správne zadaných riešení. Ak sme našli posledné správne riešenie, objaví sa bublina s nápisom ,,Môžeš prejsť na ďalšiu úlohu“. Posledné tlačidlo na obrazovke funguje podobne ako fungovalo v prvej a druhej úrovni. Po jeho stlačení prejdeme na ďalšiu úlohu len v prípade, že sme našli všetky správne riešenia. Pri tomto prípade sa spustí animácia konfiet a objaví sa gratulačná bublina. Ak sme ešte nenašli všetky správne riešenia a stlačíme toto tlačidlo, objaví sa bublina s textom ,,Najprv musíš nájsť všetky správne odpovede“. Ku každej bubline sa vždy pustí aj jej audio verzia.



Obrázok 14: Obrazovka zadaných riešení



Obrázok 15: Obrazovka druhej polovice tretej úrovne

Štvrtá úroveň disponuje oproti tretej úrovni aj dvoma kruhovými tlačidlami na pravom okraji obrazovky. Tieto tlačidlá sa objavia na obrazovke až 20 sekúnd po spustení úlohy a sú určené na zadanie riešenia úloh, ktoré nemajú riešenie alebo majú nekonečný počet riešení.



Obrázok 16: Obrazovka štvrtej úrovne

### Playground

Obrazovka playground (Obr. 17) disponuje už predtým spomenutými komponentami, no miesto pevne zadaných kameňov vidíme na obrazovke dve vrecká, jedno s horúcimi a druhé so studenými kameňmi. Z týchto vreciek vieme pomocou drag and drop funkcie kamene potiahnuť do kotlíka.



Obrázok 17: Obrazovka playground

## Editory úloh

Pre každú úroveň je navrhnutý aj editor úloh, teda priestor pre hráča navrhnúť si vlastnú úlohu. Je to zároveň aj dobrou pomôckou pre pedagógov, ktorí si vedia pre žiakov vytvoriť úlohu prispôsobenú preberanej látke. Taktiež si žiaci vedia vytvoriť úlohy pre svojich spolužiakov. Editor je dostupný v každej úrovni až vtedy, ak sme vyriešili všetky úlohy danej úrovne. Pri každom editore si vieme vypočuť vysvetlenie narábania s editorom stlačením tlačidla s informáciami.

**Editor prvej úrovne**

Pri editore prvej úrovne si vieme vybrať počiatočnú teplotu pomocou teplomera, ktorého teplotu vieme podľa potreby posúvať. Výsledná teplota, teda výsledok danej úlohy, je zobrazený v strede obrazovky a mení sa podľa zadanej teploty a zvolených kameňov. Kamene si vieme navoliť a presunúť na panel v spodnej časti obrazovky. Ďalej treba zadanie skontrolovať a ak je korektné, vieme prejsť na samotnú inicializáciu úlohy. Maximálny počet možných zadaných kameňov je osem.



Obrázok 18: Editor prvej úrovne

**Editor druhej úrovne**

Editor druhej úrovne funguje podobne ako editor prvej, až na to, že vieme pridať na panel aj znamienka a teda vytvoriť typ rovnice nachádzajúcej sa v druhej úrovni. Znamienka a kamene vieme pridávať postupne, teda na začiatku vieme pridať len znamienka, a až keď pridáme znamienko, tak vieme pridať kameň a podobne. Kamene a znamienka vieme z editora odoberať postupne, od posledného pridaného prvku. Opäť si vieme úlohu skontrolovať, a ak bude korektná, tak ju vieme inicializovať.



Obrázok 19: Editor druhej úrovne

**Editor tretej úrovne**

Pri tomto editore vieme opäť ovládať teplomer, kde nastavíme teplotu elixíru. Ďalej vieme potiahnuť dva kamene do dvoch slotov na spodnej časti obrazovky. Akonáhle obsadíme oba sloty, vieme skontrolovať správnosť zadaných parametrov. Po skontrolovaní sa objaví aj biela šípka, ktorá zobrazuje výslednú požadovanú teplotu. Taktiež sa v strede obrazovky zobrazí počet správnych riešení a farba textu sa zafarbí na zeleno. Ak neexistuje riešenie zadaných parametrov, tak sa text zafarbí na červeno. Ak má zadaná úloha riešenie, vieme si pozrieť všetky správne riešenia stlačením tlačidla na zobrazenie zadaných riešení.



Obrázok 20: Editor tretej úrovne

**Editor štvrtej úrovne**

Editor štvrtej úrovne funguje rovnako ako editor tretej úrovne. K dispozícii má aj záporné kamene a teda vieme dosiahnuť riešenia, ktoré majú nekonečný počet riešení alebo nemajú žiadne riešenie. Ak po skontrolovaní úlohy dostaneme riešenie, ktoré má buď nekonečne veľa riešení alebo žiadne riešenie, zafarbí sa príslušná ikona v strede obrazovky na zeleno.



Obrázok 21: Editor štvrtej úrovne

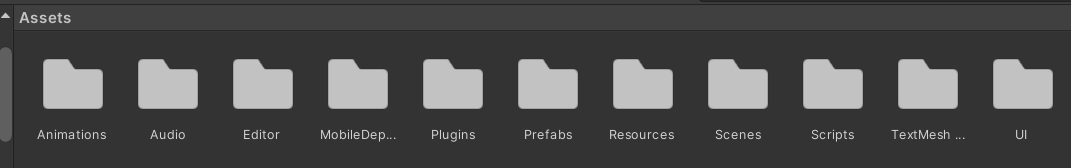
# Implementácia

V tejto kapitole prejdem samotnú implementáciu aplikácie. Prejdem aj cez štruktúru aplikácie, implementáciu jednotlivých úrovní, ovládanie aplikácie a riešenie animácií a zvuku.

Aplikácia je vyvíjaná v multiplatformovom hernom engine Unity. Pri práci s Unity si vie používateľ vybrať z dvoch programovacích jazykov, C# (vyslovuje sa sí-šarp) alebo JavaScript. Ja som si vybrala C#, keďže som s ním mala viac skúseností a väčšina online kurzov bola dostupná práve v tomto jazyku. To, čo robí Unity atraktívnym vývojovým prostredím, je jeho flexibilita a jednoduchý začiatok tvorby pre neskúsených vývojárov. Pre začiatočníkov sú pripravené online kurzy, ktoré vysvetlia základy práce s enginom. Presne pre to som si ho vybrala na tvorbu mojej práce.

## Štruktúra aplikácie

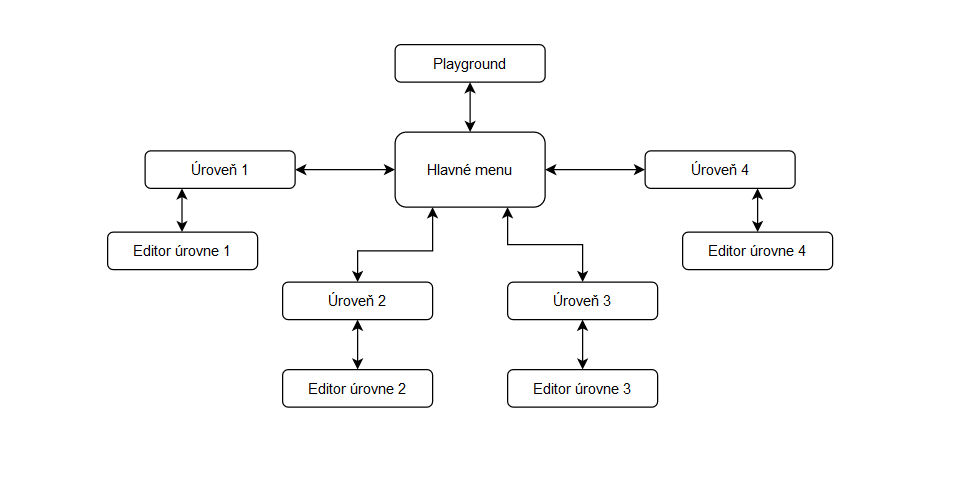
Celá štruktúra aplikácie sa nachádza v priečinku ,,Assets“. Tento priečinok je hlavným úložiskom všetkých prostriedkov, ktoré naša aplikácia použije. Nachádzajú sa tu všetky scény, skripty, animácie, preddefinované objekty (ďalej už len *prefabs*) a tak ďalej.



Obrázok 22: Assets folder

Základným stavebným blokom každej aplikácie v Unity sú scény. Každá scéna sa skladá s rôznych prvkov, ako napríklad herné objekty, plátno (ďalej už len canvas), svetlá, kamera a iné vizuálne prvky alebo objekty. Tieto prvky sú reprezentované v takzvanej hierarchii a vieme ich editovať, pridávať a odstraňovať. V mojej aplikácii sa nachádza desať scén. Na obrázku 23 vidíme prechody medzi jednotlivými scénami. Hlavné menu je scéna, ktorú vidí hráč ako prvú po zapnutí aplikácie.

Každá scéna v mojej aplikácii obsahuje herný objekt canvas, ktorý slúži na vykreslenie užívateľského rozhrania. Bez neho by sme nevideli na scéne tlačidlá, text a ďalšie komponenty spojené s užívateľským rozhraním. Canvas má v Unity niekoľko vlastností a parametrov, ktoré umožňujú prispôsobenie vzhľadu a správania prvkom užívateľského rozhrania. Medzi tieto vlastnosti patria napríklad jeho rozmer, pozícia a veľkosť prvkov, farebné schémy a mnoho ďalších. Nastavuje sa pri ňom aj responzivita aplikácie a je rodičovským objektom v hierarchii každej scény.



Obrázok 23: Prechod medzi scénami (scene flow chart)

Na canvas potom vieme postupne vkladať ďalšie herné objekty, napríklad tlačidlá alebo obrázky. Každý herný objekt existuje iba v scéne, kde bol vytvorený, teda nie je ho možné použiť v iných scénach. Pre prípad potreby herného objektu, ktorý používame viackrát, existuje možnosť vytvorenia takzvaného preddefinovaného objektu – teda prefab. Prefab funguje ako šablóna herného objektu, obsahuje všetky jeho komponenty, nastavenia a funkčnosť. Ak zmeníme obyčajný herný objekt v scéne, tieto zmeny sa neprejavia na ostatných objektoch tohto typu, no keď použijeme prefab, tak sa tieto zmeny uložia všade. V mojej aplikácii sa vo väčšine vyskytujú práve prefabs, keďže väčšina mojich objektov sa opakuje v každej scéne.

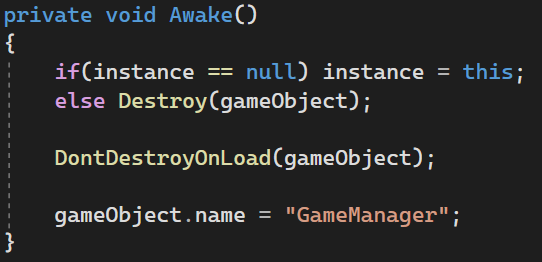
Na jednotlivé herné objekty vieme pripnúť skripty. Každý skript, v mojej aplikácii napísaný v C#, vie riadiť správania herných objektov alebo celkové riadenie aplikácie.

## Implementácia úrovní

Každá úroveň predstavuje novú scénu. Ako som už spomínala, scéna sa skladá z kamery, canvasu a herných objektov. Medzi herné objekty však patria aj objekty, ktoré nie sú vykreslené na obrazovke, no spravujú napríklad logiku danej scény. Medzi takéto objekty patria v mojej aplikácii napríklad *Game Manager*.

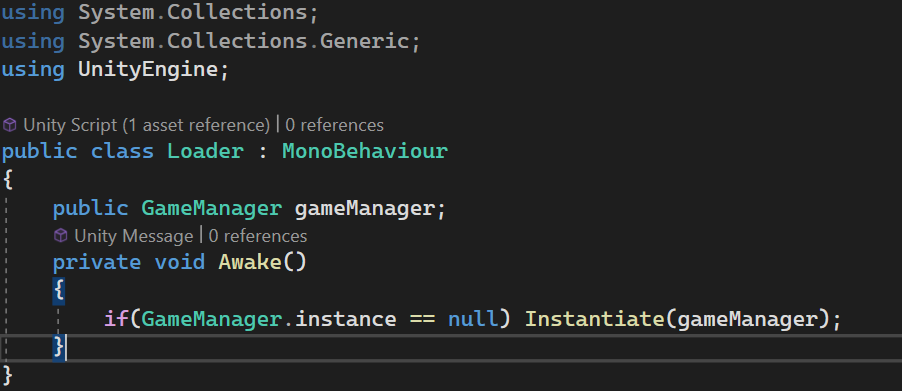
**Game Manager**

Game Manager je hlavným objektom v celej aplikácii. Samotný herný objekt má na seba pripnuté skripty: *GameManager, BoardManager, AudioManager, PlayerStats* a všetky generátory. *GameManager* skript ovláda všetky prvky aplikácie a všetky skripty, ktoré potrebujú vzájomne komunikovať, komunikujú cez neho. Toto je dosiahnuté vďaka tomu, že herný objekt Game Manager je singleton, teda existuje len jedna jeho inštancia. Zároveň pomocou *DontDestroyOnLoad* (Obr. 24) funkcie je zabezpečené, že sa herný objekt vie pohybovať medzi scénami bez toho, aby bol odstránený.



Obrázok 24: Zabezpečenie perzistencie Game Managera

Do každej scény je načítaný formou pomocného skriptu *Loader* (Obr. 25), ktorý je v každej scéne pripnutý na kameru. Samotný skript *GameManager* zabezpečuje napríklad načítanie plochy pomocou skriptu *BoardManager*, kontrolu zadaného riešenia úloh v každej úrovni, kontrolu počtu nájdených riešení, zobrazovanie bublín v každej úrovni, zmeny farieb, obrázkov a animácií.



Obrázok 25: Kód triedy Loader

**Board Manager**

Skript *BoardManager* zabezpečuje všetko, čo sa deje na samotnej ploche. Teda konkrétne nastavenie jednotlivých úrovní, teplomera, inicializácia úloh a kameňov na ploche podľa príslušného generátora. Najdôležitejšou metódou je metóda *SetUpBoard*, ktorá podľa príslušnej úrovne najprv skontroluje, či sa nachádzajú kamene na ploche, ak áno, odstráni ich. Ďalej kontroluje, či máme uloženú nedohranú úlohu, ak áno, následne inicializuje uloženú úlohu. Ak nie, tak si vypýta informácie od príslušného generátora a inicializuje úlohu. Pri inicializovaní novej úlohy si vypýta dodatočné informácie od metódy *GetTaskBounds*, ktorá vráti požadovanú gradáciu, ktorú potom pošle generátoru a na základe tejto informácie vytvorí generátor úlohu s príslušnou gradáciou.

**Perzistencia informácií medzi scénami**

Ukladanie informácií medzi scénami som vyriešila pomocou skriptu *PlayerStats,* ktorý slúži na ukladanie informácií o každej úrovni. Napríklad informácie ako: aké kamene boli na ploche, aká bola výsledná teplota, aká bola hodnota na teplomere a tak ďalej. Tieto informácie si uchováva o každej úrovni a *BoardManager* si ich vypýta, ak potrebuje načítať nedokončenú úlohu. Ukladá si aj progres každej úlohy a na základe neho automaticky otvorí ďalšiu úroveň alebo editor úloh.

**Generátor prvej a druhej úrovne**

Skript generátora prvej a druhej úrovne generuje úlohy podľa zadaných parametrov, akými sú napríklad maximálny povolený počet kameňov, maximálna hodnota kameňa a či používame aj zátvorky alebo prácu so znamienkami. Pri generátore týchto úrovní sa používa pomocný skript *Generators,* ktorý generuje diofantickú rovnicu a overuje jej správnosť pomocou rozšíreného Euklidovho algoritmu. Ak máme vygenerovanú rovnicu, následne kontrolujeme, či sme ju už predtým neriešili. Ak sme ju už riešili, vygeneruje sa ďalšia a proces sa opakuje. Potom uloží generátor všetky údaje do skriptu *PlayerStats* podľa toho, či sa jedná o úlohu prvej alebo druhej úrovne*.*

**Generátor tretej a štvrtej úrovne**

Skript generátora tretej a štvrtej úrovne generuje úlohy podľa zadaných parametrov, akými sú napríklad počet dostupných kameňov, maximálna hodnota kameňa, maximálny počet možných riešení a či ide o prvý alebo druhý typ úloh v tretej úrovni. Pri generátore už nevyužívame pomocný skript, ale generujeme náhodne kamene a pomocou backtrackingu nájdeme vyhovujúce zadanie. Skript má samostatnú metódu generátora pre tretiu a štvrtú úroveň.

Pri tretej úrovni sa pozeráme na to, či ide o prvý alebo druhý typ úloh, respektíve či máme konečný alebo nekonečný počet kameňov. Ďalej sa pozeráme na to, či máme k dispozícii dva alebo tri rôzne kamene. Pri štvrtej úrovni sa náhodne rozhodneme, či ideme hľadať úlohu, ktorá má konečný alebo nekonečný počet riešení. Ďalej sa pri každej z nich opäť náhodne rozhodujeme, či hľadáme úlohu, ktorá nemá ani jedno riešenie.

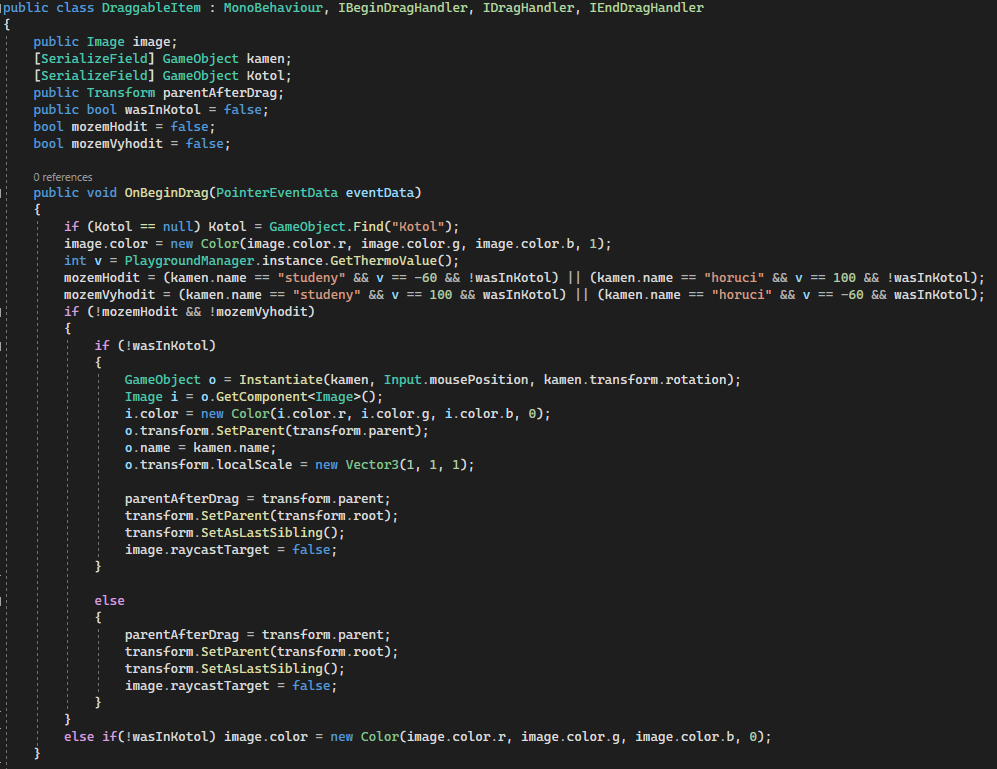
Podľa týchto parametrov hľadáme zadanie pomocou metódy *CombinationSum,* ktorá využíva metódu *Backtrack.* Po nájdení zadanie uložíme parametre do skriptu *PlayerStats*.

## Ovládanie aplikácie

**Drag and drop**

Pri práci s kameňmi v jednotlivých úrovniach sa používa funkcia drag and drop. Aby funkcia fungovala správne, bolo potrebné využiť knižnicu *UnityEngine.EventSystems*, konkrétne pri funkcii drag treba využiť triedy *IBeginDragHandler, IDragHandler, IEndDragHandler* a pri funkcii drop treba využiť triedu *IDropHandler*. Ďalej som vytvorila metódy *OnBeginDrag (Obrázok 26), OnDrag* a *OnEndDrag* pre funkciu drop a *OnDrop* metódu pre funkciu drop. *OnBeginDrag* kontroluje, či už kameň v kotlíku bol alebo nie, a podľa toho zabezpečí jeho správanie. *OnDrag* len mení pozíciu ťahaného objektu podľa pohybu na obrazovke len v prípade, ak sa kameňom smie hýbať. *OnEndDrag* rozhoduje o správaní kameňa podľa toho, kam sme sa ho snažili posunúť. Ak sme ho posunuli do kotlíka a ešte v ňom nebol, pridá ho tam a pripočíta jeho hodnotu na teplomer. Ak už kameň v kotlíku bol a snažíme sa ním hýbať, tak kameň z kotlíka vyhodí a jeho hodnotu odpočíta z teplomera. Nakoniec nastaví rodičovský objekt kameňa a nastaví mu príslušnú veľkosť.

Skript *DraggableItem* je zodpovedný za funkciu drag. Tento skript musí byť pripnutý na každý objekt, ktorý chceme, aby sa dal presúvať pomocou tejto funkcie. Skript *DropItemSlot* treba pripnúť na každý herný objekt, ktorý chceme aby fungoval ako objekt, kam môžeme presunúť objekty so skriptom *DraggableItem*. V mojej aplikácii je implementovaných viacej skriptov na funkciu drag and drop z toho dôvodu, že pri niektorých úrovniach je potrebné iné správanie presúvaného objektu alebo iné podmienky pridania do kotlíka. Preto majú tretia úroveň, editory tretej a štvrtej úrovne a playground každý svoj vlastný skript na funkciu drag and drop.



Obrázok 26: Metóda OnBeginDrag

## Animácie a zvuk

**Animácie**

V mojej aplikácii sa nachádzajú menšie animácie v podobe pohybu ohníka v pozadí alebo pod kotlíkom, miešanie elixíru a gratulačné konfety. Každá animácia má *Animation Clip* a *Animator Controller*. Na herný objekt, cez ktorý chceme animáciu ovládať, musíme pridať komponent *Animator*, kam potom zadáme požadovaný *Animator Controller*. Pri animácii konfiet alebo odomknutia novej úrovne používame *triggers*, ktoré nastavujeme podľa potreby prehratia animácie.

**Zvuk**

Zvuk v mojej aplikácii predstavuje hlas ježibaby Armidy. Ako som už bližšie spomínala, na pri každej úrovni nájdeme tlačidlo informácií, kde nám ježibaba vysvetľuje princíp úlohy. Ďalej zvuk využívame aj pri riešení úloh, kde dostávame spätnú väzbu vizuálne (pomocou bublín) aj pomocou zvuku (hlas Armidy). V Unity vieme každému hernému objektu pridať komponent *Audio Source*, ktorý slúži na ovládanie pridelenej zvukovej nahrávky pomocou skriptov. Objekt Game Manager má na sebe pripojený skript *AudioManager*, ktorý má na starosti všetku audio spätnú väzbu. Každé tlačidlo informácií má na sebe pripojený skript *InfoAudioManager*, ktorý rieši prehrávanie a zastavenie príslušne prideleného audia.

# 4. Testovanie

V tejto kapitole sa venujem cieľu a samotnému priebehu testovania aplikácie. Pre účely testovania som navštívila súkromnú základnú školu BESST v Trnave.

## 4.1 Cieľ testovania

Cieľom testovania je najmä zistiť, či sa aplikácia dá podľa detí ovládať, či je intuitívna a žiakov zaujme. Žiaci, ktorí testovali moju aplikáciu navštevujú štvrtý ročník na základnej škole, kde sa už oboznámili s prostredím Ježibaba Armida. Teda cieľom nebolo im vysvetliť toto prostredie, ale testovať, či vedia s aplikáciou pracovať na základe ich predošlých poznatkov o danom prostredí.

## 4.2 Priebeh testovania

Testovanie prebehlo 20. apríla 2023, kedy som mala vyhradenú jednu celú vyučovaciu hodinu, teda 45min a aplikáciu testovali všetci žiaci štvrtej triedy. Všetci žiaci majú vlastné tablety s operačným systémom IOS a aplikácia bežala na webe pomocou webGL. Na začiatku hodiny si všetci vybrali tablety a s pomocou učiteľa sme im poslali webovú adresu, na ktorej si zapli aplikáciu. Zo začiatku sme im museli pomôcť s nastavením tabletov, potom som im vysvetlila základný princíp ovládania prvej úrovne, no ani som to nemusela hovoriť všetkým, pretože niektoré deti na to prišli samé a už sa pustili rovno do práce. Nechali sme ich samých pracovať, no postupne sa zdvíhali ruky a pýtali sa ma otázky ohľadom prvej úrovne, teda asi bola pre nich moc zložitá. Neskôr si vytiahli papiere a začali si počítať na nich, tam som sa opäť utvrdila v tom, že sú asi úlohy v prvej úrovni moc komplikované. Jedna žiačka mi povedala, že by si rada krokovala postupne jednotlivé kamene, no môj komponent pre zadanie výslednej teploty jej to nedovolil, pretože je nastavený na určité rozmedzie okolo výslednej teploty. Vďaka tomuto som si uvedomila, že si to asi viacej detí bude chcieť krokovať, takže zmením aj tento komponent.

Niektoré deti prešli pomerne rýchlo cez prvú úroveň a niektorým to zabralo väčšinu hodiny. Keď už som videla, že sa s tým trápia, povedala som im, nech sa pokojne presunú na ďalšiu úroveň, vtedy boli nadšené, pretože sa im už nechcelo počítať. V druhej úrovni narazili na ďalší kameň úrazu, a to odčítanie záporných čísiel a práca so zátvorkami. Postupne som k nim chodila a pomáhala im, dokonca som si spomenula na moju prvú návštevu školy, kde som sa bola pozrieť na ukážkovej hodine. Na tejto hodine mali taktiež problém s pochopením odčítania záporných čísiel a vtedy jedna žiačka povedala veľmi múdru myšlienku: *,,Predstavte si, že máte na sebe studenú mikinu. Ak si dáte dole studenú mikinu, tak vám bude teplo. Preto keď odčítame záporné číslo, dostaneme kladné číslo.“* Pomocou tejto myšlienky som im vysvetlila ako to funguje ak majú odrátať záporné číslo a hneď to pochopili a tešili sa z ďalších úloh. Pri práci so zátvorkami som im len pripomenula, že mínus pred zátvorkou mení znamienko kameňov v zátvorke a pekne riešili ďalej.

Nasledovala tretia úroveň, ktorá bola celkovo najúspešnejšia. Žiaci od začiatku skúšali prísť na systém práce s komponentami v tejto úrovni, no hneď sa chytili a začali pracovať. Pri tejto úrovni nemali žiadne problémy, dostali sa aj na druhú časť úrovne, kde už pracujú aj so zápornými kameňmi a majú ich len obmedzený počet. Už počas testovania mi hovorili, že táto úroveň je ich najobľúbenejšia a že chcú viacej takýchto úloh.

K štvrtej úrovni sa dostali len dvaja žiaci. Jeden z nich za mnou prišiel s tým, že už vyriešil všetky tri možné riešenia, ale úloha ho nepustí ďalej. Upozornila som ho na to, že v štvrtej úrovni pribudli dve tlačidlá na riešenie úloh s nekonečným alebo nulovým počtom riešení. Tie si nevšimol, takže aj tie budem musieť zmeniť, alebo na to viac upozorniť.

## 4.3 Vyhodnotenie testovania

Celkovo hodnotím testovanie ako pozitívne. Žiakom sa aplikácia veľmi páčila a po skončení hodiny za mnou prišli a hovorili, že tretia úroveň bola super a že tam chcú viacej úloh. Páčili sa im aj animácie na pozadí a celkovo grafika aplikácie. Pre mňa bolo najviac prínosné vidieť to, ako sa im aplikácia veľmi páčila a robenie úloh ich úprimne bavilo a samozrejme aj zistenie, že prvá úroveň je pre nich náročnejšia a že niektoré deti by si to chceli krokovať a teda musím upraviť komponent na zadanie výslednej teploty

Chcela by som ešte raz poďakovať pedagógom, ktorí mi pomáhali pri testovaní aplikácie a hlavne samotným deťom, ktoré mi pomohli dotiahnuť prácu podľa ich pripomienok.



Obrázok 27: Testovanie aplikácie



Obrázok 28: Testovanie aplikácie

**Záver**

# 5. Zdroje

[1] Benefits of Educational Software for Young Learners. Dostupné na internete: <https://essentialskills.com/blog/benefits-of-educational-software-for-young-learners>

[2] Hejného metóda: Zaslúžená radosť z poznávania. Dostupné na internete: <https://www.h-mat.cz/hejneho-metoda>

[3] Hejného metóda: 12 kľúčových princípov. Dostupné na internete: <https://www.h-mat.cz/principy>

[4] Didaktické prostredie Ježibaba Armida. Dostupné na internete: <https://blog.h-mat.cz/didakticka-prostredi/armida>

[5] Math Slide: add & subtract. Dostupné na internete: <https://apps.apple.com/us/app/math-slide-add-subtract/id588603890>

[6] Matemág: vzdelávacia hra. Dostupné na internete: <https://matemag.sk/>

[7] BARILLOVÁ, C. Softvérová podpora vyučovania matematiky Hejného metódou – Prostredie Krokovanie. Bratislava: Fakulta matematiky, fyziky a informatiky, UK 2022. 43s

[8] STAVKOVA,L.2018.Diofantos, diofantické rovnice a jejich soustavy. Brno: Katedra matematiky, PED MUNI 2018. 58s

Dostupné na: <https://is.muni.cz/th/lv5ac/BP\_Lucie\_Stavkova.pdf>

[9] OLŠIAKOVA,V.2009.Euclides z Alexandrie. Ružomberok: KU 2008/2009. 7s

Dostupné na: <http://math.ku.sk/~tkacik/predmety/download/hm/prace/olsiakova.pdf>

[10] VAŠULÍNOVÁ,M.2016.Diofantické rovnice a jejich užití. Brno: Katedra matematiky, PED MUNI 2016. 54s

Dostupné na: <https://is.muni.cz/th/vf3oi/Diofanticke\_rovnice\_a\_jejich\_uziti.pdf>

[11] MORKÁ,T.2015.Lineární diofantické rovnice. Brno: Katedra matematiky, PED MUNI 2015. 53s

Dostupné na: <https://is.muni.cz/th/prfye/bakalarska\_prace.pdf>

[12] STEINSDORFER,J.2015.Metody řešení diofantických rovnic. Ústí nad Labem: Prírodovedecká fakulta, Univerzita Jana Evangelisty Purkyně 2015. 99s

Dostupné na: <http://physics.ujep.cz/~jmaly/mo/Diofanticke\_rovnice.pdf>

[13] PORUBSKÝ, Štefan. ROZHLEDY matematicko-fyzikálni, ročník 64. MÚ SAV Bratislava, 1985-86, 5s. Dostupné na: <http://jacobi.cs.cas.cz/pdf/P1985RT.pdf>

[14] HEJNÝ, M a spol. Matematika pre 4. ročník základnej školy. Indícia, s.r.o. Bratislava, 2021, 72s

1. Diophantus’s riddle <https://mathworld.wolfram.com/DiophantussRiddle.html> [↑](#footnote-ref-1)
2. Hilbert’s problems <https://mathworld.wolfram.com/HilbertsProblems.html> [↑](#footnote-ref-2)
3. Rozšírený Euklidov algoritmus <https://www.extendedeuclideanalgorithm.com/xea.php> [↑](#footnote-ref-3)
4. Backtracking <https://jeffe.cs.illinois.edu/teaching/algorithms/book/02-backtracking.pdf> [↑](#footnote-ref-4)