

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

Softvérová podpora vyučovania matematiky Hejného metódou –
prostredie pavučiny

Bakalárska práca

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

Softvérová podpora vyučovania matematiky Hejného metódou –
prostredie pavučiny

Bakalárska práca

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



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Tatiana Gyurcsovicsová
Študijný program: aplikovaná informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický
Názov: Softvérová podpora vyučovania matematiky Hejného metódou - prostredie Pavučiny
*Educational software for Hejny's method of mathematics teaching environment
Spider Network*

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 prostredia 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 gradujujúcich úloh/úrovní zodpovedajúcich konceptom, ktoré žiaci na danej úrovni objavujú. Aplikácia precvičuje každú úlohu/úroveň na sade predvolených a generovaných zadaní. Až po jej zvládnutí môže žiak pokročiť do ďalšej úrovne. Žiak má možnosť vytvoriť vlastné zadanie v rámci každej úlohy/úrovne. Pri návrhu nového zadania (ako aj pri jeho riešení) aplikácia indikuje počet existujúcich/zostávajúcich riešení daného zadania. Generátor zadaní musí generovať zadania s rozumným počtom existujúcich riešení. Aplikácia si ukladá výsledky práce žiaka, ponúka možnosť priebežnej kontroly a prehľad hodnotenia úspešnosti. Prvé testovanie s deťmi v triede sa predpokladá v apríli, druhé testovanie v triede sa predpokladá v júni.

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



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

Dátum zadania: 02.10.2019

Dátum schválenia: 14.10.2019

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

.....
š student

.....
vedúci práce

Pod'akovanie

Chcem sa pod'akovať svojmu školiteľovi RNDr. Petrovi Borovanskému, PhD. za cenné rady, nekonečnú trpezlivosť a čas, ktorý mi vevonal počas konzultácií a písania bakalárskej práce. Pod'akovanie patrí aj RNDr. Dagmar Môt'ovskej, PhD., ktorá mi umožnila lepšie porozumieť princípom Hejného metódy počas vizitácie na Súkromnej základnej škole pre žiakov so všeobecným intelektovým nadaním v Bratislave, Bajkalská 10. V neposlednom rade ďakujem svojej rodine, ktorá ma podporovala počas celého môjho štúdia.

Abstrakt

Abstract

Obsah

Úvod

Za posledné desaťročie sa mobilné zariadenia stali neodmysliteľnou súčasťou našich životov. Počítače, tablety a mobilné telefóny sú dostupnými pomocníkmi, ktorých využívame každý deň. Tomuto faktoru sa prispôsobuje aj vyučovací proces. Na školách sa čoraz častejšie používajú moderné technológie. Tie zohrávajú dôležitú úlohu napríklad pri využití edukačného softvéru na vyučovacích hodinách alebo pri zadávaní domácich úloh v elektronickej podobe. Tento spôsob vzdelávania sa osvedčil aj počas pandémie vírusu COVID-19. Pár hodín po vyhlásení zatvorenia škôl sa celý vyučovací proces podarilo preniesť do online podoby. Toto bolo možné vďaka dostupnosti zariadení a edukačného softvéru.

Cieľom tejto bakalárskej práce je vytvoriť mobilnú aplikáciu podľa princípov Hejního metódy výučby matematiky. Túžba poznávať je prirodzenou ľudskou vlastnosťou, ktorá je u dieťaťa niekoľkonásobne väčšia ako u dospelého. Miesto, kde sa dieťa dostane k informáciám jednoducho, je škola. Mnohé deti vnímajú túto inštitúciu ako trest. Škola má byť pre dieťa miestom, kde objavuje nové poznatky s radosťou a nadšením. Veselé prostredie aplikácie učí dieťa matematickým princípom tak, ako je dieťaťu prirodzené učiť sa. Aplikácia má byť vhodným rozšírením učiteľovho portfólia.

Je vhodná pre všetkých žiakov. V prípade, ak dieťa rýchlejšie pochopí učivo a úlohy ho baví, učebnica a pracovný zošit mu nemusia stačiť. Pomalšie dieťa môže naopak potrebovať viac príkladov na pochopenie princípov, ako poskytuje učebnica, či pracovný zošit.

Naša mobilná aplikácia si kladie za cieľ pomôcť pri spôsobe výučby s využitím moderných technológií. Zaoberať sa bude prostredím pavučiny.

Prostredie pavučín pracuje s grafmi. Graf dieťa vníma ako pavučinu, na ktorej sa učí pracovať s jednoduchými matematickými operáciami. Pohybom po hranách pavučiny si precvičuje jednoduché matematické operácie sčítavanie a odčítavanie. Skladaním viacerých hrán sa podvedome učí rozdeľovať čísla na diely.

Hejného metóda podporuje prácu v skupinách, čím motivuje deti navzájom komunikovať. Aplikácia je primárne vytvorená pre jednotlivca a preto ju nemôžeme považovať za nástroj na výučbu prostredia pavučín pomocou Hejného metódy. Aplikáciu sme napriek tomu vytvárali s motiváciou priniesť nové, neobjavené úlohy deťom, ktoré

prostredie pavučiny poznajú, ale aj tým, pre ktoré je toto prostredie, či Hejného metóda, neznámou. Rovnako dúfame, že aplikácia by mohla byť nápomocná v prípade situácií, keď nebude možné učiť sa v škole. Veríme, že aj za pomoci tejto aplikácie sa Hejného metóda bude tešiť väčšiemu záujmu verejnosti.

1. Východisková kapitola

V tejto kapitole si priblížime pedagogický softvér s kľúčovými požiadavkami, ktoré by mala spĺňať aplikácia určená na vzdelávanie detí. Bez pochopenia zmyslu a cieľov Hejného metódy nie sme schopní kvalitne spracovať jedno z jej prostredí. Preto venujeme jej viac pozornosti. Priblížime si prostredie pavučiny, ktorému sa naša práca venuje, a jeho spôsob zaradenia do výučby na školách. A pozrieme sa na iné riešenia, ktoré sú založené na Hejného metóde, alebo sa ňou inšpirovali. V poslednej časti si ozrejmime použité technológie pri vytváraní našej práce.

1.1. Pedagogický softvér

Medzi pedagogický softvér zaraďujeme softvérové prostriedky, ktoré sa cieľavedome používajú na podporu učenia a učenia sa [1]. Ďalší zaujímavý pohľad na pedagogický softvér je - Na kvalitný edukačný softvér sa musí dať pozeráť ako na múdry „papier“ (na ploche obrazovky), ktorý nám pomáha experimentovať, manipulovať s objektami, aktívne objavovať vzťahy a zákonitosti, skúmať a konštruovať ich [1]. O edukačnom softvéri môžeme teda povedať, že sú to také prostriedky, ktoré sú určené pre učiteľa ako pomoc pri výučbe, no zároveň i také, ktoré sú určené pre žiakov ako pomoc pri samovzdelávaní. Keďže sa edukačný softvér na školách využíva stále viac, je nutné naň kladť vysoké nároky. Je taktiež veľmi dôležité uvedomiť si, že pri softvéri, ktorého primárnym používateľom je dieťa, je používateľské rozhranie rovnako dôležité ako samotný obsah.

Oboznámme sa so základnými požiadavkami, ktoré by podľa [1] mal edukačný softvér spĺňať. Mal by:

- mať používateľské rozhranie primerané veku adresáta a účelu používania
- používať vizuálne manipulovateľné objekty, vizualizáciu údajov, stavu práce
- byť interaktívny, reagovať na riadenie a naše požiadavky, rovnako i spolupracovať s používateľom
- byť otvorený voči novej grafike, aktivitám, vlastným zariadeniam
- podporovať náš didaktický zámer(napr. poskytovať spätnú väzbu)
- podporovať rozmanitosť, atraktivnosť a pestrosť aktivít
- byť koncentrovaný na danú tému a daný účel, čiže nezaťažovať používateľa inými problémami a teda nerozptyľovať tak pozornosť

- poskytovať úroveň náročnosti a podporovať individuálny prístup žiaka

Pri tvorbe, ako aj používaní pedagogického softvéru je potrebné mať na pamäti, aký je konkrétny účel tohto softvéru – ako má deťom pomôcť pri poznávaní a čoho. Treba tiež dokázať využiť jeho potenciál naplno. Správnym spracovaním náučného obsahu do interaktívnejšej formy sa dá zvýšiť motivácia žiakov, umožniť im získať konkrétne skúsenosti a utužiť vytvorené poznatky [1].

1.2. Hejného metóda

Za otca tejto netradičnej metódy výučby matematiky sa považuje český matematik a pedagóg Vít Hejný. Otcovi Hejného metódy sa nepáčilo k čomu viedli a nabádali matematické úlohy, presnejšie sa mu nepáčilo memorovanie vzorcov, namiesto pochopenia problému. Tieto vzorce totiž väčšinou stačili len na riešenie štandardných problémov, no zložitejšie príklady žiaci neboli schopní riešiť. Toto ho viedlo k vytváraniu nového typu úloh, ktoré nechal riešiť svojich žiakov rovnako ako i svojho syna. Zámerom týchto úloh nebolo donútiť žiaka memorovať, ale poskytovali čas na analyzovanie problému, nájdenie vlastného spôsobu riešenia a následne diskutovať o získaní riešenia. Neskôr práve jeho syn Prof. RNDr. Milan Hejný, CSc túto metódu výučby rozvinul ďalej spolu so svojimi spolupracovníkmi. Hejného metóda sa prezentuje tým, že podporuje kreativitu a samostatnosť. Taktiež nevníma chyby ako niečo neprípustné, práve naopak vidí chyby ako jeden z prostriedkov zlepšovania sa. Často využívaný pojem v tejto metóde je neriešiteľnosť úlohy, pomocou neho taktiež rozvíja myslenie dieťaťa. V dnešnej dobe je táto metóda úspešne rozšírená vo svete a nie je prekvapením, že sa stále teší väčšej obľube. Jej ucelenú podobu publikoval až Prof. RNDr. Milan Hejný, CSc, aj vďaka ktorému sa dnes môžeme oboznámiť s dvanástimi kľúčovými princípmi Hejného metódy [2]:

- Budovanie schém – dieťa vie aj to, čo sme ho nenaučili
Schéma je súhrn navzájom prepojených znalostí so známeho prostredia. Pre človeka je prirodzené vytváranie a budovanie pomyselných schém. Podvedome si schémy vytvárame na základe skúseností s prostredím a využívame ich ako hlavný nástroj rozhodovania. Veľmi podobne fungujú aj matematické schémy. Dôležité je však správne podnietiť ich vývoj. Dôležitým faktorom je, aby deti schému potrebovali k vyriešeniu úlohy a zároveň aby sa nové poznatky, ktoré chceme, aby do schémy zapadli,

objavili vo vhodnej chvíli. Pre vznik a rozvoj pevných základov sa snaží Hejného metóda budovať takéto schémy.

- Práca v prostrediach – učíme sa opakovanou návštevou
Hejného metóda vo svojimi približne dvadsiatimi-piatimi rôzne fungujúcimi prostrediami snaží dokázať to, že ak dieťa pozná prostredie, v ktorom sa cíti dobre, nerozptyľujú ho neznáme veci. Vďaka tomu, že prostredia sú dieťaťu známe, môže vnímať učenie ako hru. V každom prostredí nájdeme rôzne zložité úlohy, v ktorých sa náročnosť problému zväčšuje a zároveň môže byť prispôsobená pre každé dieťa individuálne.
- Prelínanie tém – matematické zákony neizolujeme
Ak sa učíme jednotlivé fakty, či pravidlá izolovane bez toho aby sme ich skutočne pochopili, nemusíme si na tieto fakty časom spomenúť. To súvisí s tým, že človek bez nájdania zmyslu úlohy, nie je motivovaný niečo urobiť. I práve preto je dieťaťu nepríťažlivé sústrediť sa na jednu tému. O to príťažlivejšie sú problémy, v ktorých po ceste k riešeniu dieťa spozná viacero tém, pochopí ich dôležitosť ale i využitie. Tým že dieťa v Hejného metóde objavuje samo a rovnako sa samo i rozhoduje, lepšie si zapamätá čo využilo a ako to využilo. Takto sa postup riešenia stáva súčasťou schémy, v ktorej sa dieťaťu dobre hľadá, ak z nej niečo potrebuje opätovne využiť.
- Rozvoj osobnosti – podporujeme samostatné uvažovanie dieťaťa
Škola nie je len o získavaní nových vedomostí z vyučovacích predmetov, ale i o budovaní a rozvíjaní pracovných a sociálnych návykov. Napriek tomu, že matematika je prevažne o faktoch, diskusia pri jej výučbe má svoj význam. Pre zapojenie sa do diskusie je nutné premýšľať nad problémom, hľadať a chápať argumenty. Hejného metóda týmto spôsobom učí, že je nutné načúvať druhým, že väčšina nemusí mať vždy pravdu a rovnako aj to, že spraviť chybu nie je koniec sveta, práve naopak, chyba dáva priestor na poučenie sa z nej.
- Skutočná motivácia – keď „neviem“ ale „chcem vedieť“

Motivácia dáva procesu poznávania energiu aj orientáciu, práve preto hrá kľúčovú rolu v kvalite vyučovacieho procesu. Ak má dieťa vnútornú potrebu poznávať, poznáva intenzívnejšie, hlbšie a zároveň komplexnejšie ako dieťa, ktoré bolo k procesu poznávania donútené. Hejného metóda udržiava záujem a motiváciu využívaním známych prostredí, pomocou ktorých dieťa spoznáva abstraktné pojmy, bez toho aby tieto pojmy dieťa vnímalo ako niečo nové a cudzie, ale ako rozšírenie doterajších poznatkov.

- Reálne skúsenosti – stavíme na vlastných zážitkoch dieťaťa
Hejného metóda vychádza predovšetkým z vlastných skúseností dieťaťa. Vlastné skúsenosti nie je možné dieťaťu predať, musí ich získať samo. V matematike sa to odzrkadľuje nasledovne – aby dieťa porozumelo fungovaniu vecí, musí riešiť úlohy. Aktívnym premýšľaním o problémoch si dieťa niečo odnáša, pričom nezáleží na správnosti výsledku. Z nesprávneho výsledku si zapamätá, kadiaľ cesta nevedie, poprípade zistí čo za poznatky mu chýbajú.
- Radosť z matematiky – výrazne pomáha pri ďalšej výuke
Skúsenosť učiteľov hovorí, že najúčinnějšía motivácia prichádza s pocitu úspechu. Práve preto musia byť úlohy navrhnuté spôsobom, ktorý dieťa nenudí a zároveň boli vyriešiteľné pomocou vynaloženej námahy dieťaťa. Týmto spôsobom má dieťa dobrý pocit z odvedenej práce. Radosť z učenia sa zväčší aj keď dieťa pri riešení problému objaví niečo nové.
- Vlastný poznatok - má väčšiu váhu ako ten prevzatý
Hejného metóda na rozdiel od klasických metód vyučovania, pri výučbe prechádza od skúsenosti k zavedeniu pojmu. Metóda vedie dieťa k tomu aby pochopilo pojem a jeho fungovanie samo a zároveň aby sa samo rozhodlo v akej situácii je vhodné pojem použiť. Najprv teda príde pochopenie, nasledovne sa mu priradí názov na základe potreby vyjadrovania sa o tomto poznatku tak, ako je človeku prirodzené.
- Rola učiteľa – sprievodca a moderátor diskusií

Úlohou učiteľa Hejného metódy nie je predávať svoje poznatky, tak ako to robí učiteľ vyučujúci klasickou výučbovou metódou. Na rozdiel od takéhoto učiteľa, vytvára priestor na to, aby dieťa získalo poznatky samo. Jeho úlohou je taktiež vhodne zvoliť úlohy pre rôzne pokročilé skupiny žiakov v rámci jednej triedy. Namiesto prezradenia riešenia, otvára a zároveň podnecuje diskusiu o riešení úlohy. Riešenie hľadá dieťa a následne svoje riešenie prezentuje nielen učiteľovi ale i spolužiakom. Taktiež učiteľ neopravuje chyby, jeho úloha spočíva v nabádaní dieťaťa v ich hľadaní, i keď je riešenie momentálne správne.

- Práca s chybou – predchádzame zbytočnému strachu u detí
Ak má dieťa strach z chyby, môže to spôsobiť jeho otáľanie pri riešení úlohy a čakanie na výsledok, ku ktorému dospeje zvyšok kolektívu. Pri takomto prístupe sa stráca všetko to, čo Hejného metóda podporuje (motivácia, radosť z učenia sa, získavanie skúseností). No pri správnom prístupe sa z chyby dá ťažiť. Dôležitejšia je analýza chyby ako samotný proces hľadania chyby. Dôležité je vysvetlenie ako chyba vznikla, či už chybu spravilo dieťa, alebo učiteľ. Týmto spôsobom môžu všetci žiaci chybe porozumieť. Práve preto sú chyby v Hejného metóde vítane.
- Primerané výzvy – pre každé dieťa zvlášť, podľa jeho úrovne
Rozloženie schopností detí v hocijakej triede nikdy nie je rovnomerné. Nevhodné množstvo obtiažnych úloh demotivuje slabšie dieťa, a naopak priveľa jednoduchých úloh nudí nadanejšie dieťa. Cieľom je obsiahnuť do vyučovacej hodiny príklady s takou náročnosťou, ktoré dokáže vyriešiť samo aj najslabšie dieťa v kolektíve, no zároveň aj príklady, ktoré vyrieši najlepší žiak, poprípade nikto. Výberom úloh sa samozrejme zaoberá učiteľ, ktorého úlohou je aj chváliť úspech, či chváliť originalitu myšlienok dieťaťa, pri čom tieto myšlienky môžu byť aj mylné.
- Podpora spolupráce – poznatky sa rodia vďaka diskusiám
Hejného metóda nepodporuje vznik bariér medzi spolužiakmi kvôli strachu z trestu, ktorý hrozí pri opisovaní alebo spolupráci pri riešení úloh. Každé dieťa sa potrebuje podeliť o svoje názory a poznatky. Preto aj po nájdení

správneho výsledku stále ostáva priestor aj pre ten chybný, ten totiž podnecuje dieťa, ktoré prišlo k správne výsledku, k hlbšiemu zamysleniu sa nad postupom, ktorý zvolili a odôvodneniu správnosti svojho výsledku. Dieťa však stále môže pracovať samostatne, ak mu to tak vyhovuje.

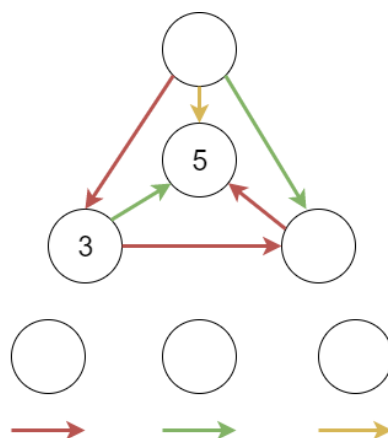
Možnosť pozorovať hodinu matematiky odučenú Hejného metódou sa mi naskytla počas mojej vizitácie na Súkromnej základnej škole pre žiakov so všeobecným intelektovým nadaním v Bratislave, Bajkalská 10. Na prvý pohľad sa hodina zdala byť príliš chaotická a hlučná na to, aby sa v takomto prostredí dieťa bolo schopné niečo naučiť. Napriek tomu hodinu stále viedla učiteľka a jediné požiadanie o prácu v tichosti stačilo na to, aby sa deti utíšili. Deti aj napriek tomu, že pracovali samostatne, každý z nich riešil zadané príklady, z čoho bolo zjavné, že ich príklady bavia. Popri riešení príkladov sedeli deti za okrúhlim stolom, diskutovali o riešení so svojim susedom a ostatnými spolužiakmi, okrem iného si navzájom pomáhali nájsť a uvedomiť chyby.

1.3. Prostredie Pavučiny

Prostredie pavučiny je jedno so základných prostredí v Hejného metóde. Vo vyučovacom procese sa používa od prvého ročníka a využíva sa počas celého prvého stupňa, dokonca i na druhom stupni. Úlohy v tomto prostredí sa skladajú z dvoch vecí. Zo samotnej pavučiny, ktorá môže mať rôzny tvar, počet parametrov a tiež usporiadanie. Druhou, neodmysliteľnou súčasťou úlohy z prostredia pavučín je legenda, ktorá nám hovorí o hodnotách jednotlivých šípok použitých v pavučine v závislosti na ich farbe.

Ak pavučiny vyjadríme pomocou jazyku matematiky, sú to orientované, ohodnotené grafy. Pavučina má ohodnotené vrcholy, tieto číslo môžu byť prirodzené, vrátane nuly, no je možné uvažovať aj so zápornými číslami. Pavučina má taktiež ohodnotené orientované hrany, to znamená, že pri každej hrane je uvedená jej orientácia vyjadrená šípkou a prirodzeným číslom, tentokrát však bez nuly, ktoré je zakódované určitou farbou.

Základným stavebným kameňom každej pavučiny je číslo, ktoré predstavuje stav a šípka, ktorá pomocou svojej hodnoty, farby a orientácie reprezentuje proces, zmenu. Úloha v prostredí pavučín je ohodnoteným grafom, v ktorom sú niektoré údaje skryté a je na riešiteľovi, aby tieto údaje doplnil.



Obrázok č.1: ukážka príkladu v prostredí pavučiny

1.4. Podobné existujúce riešenia

Hejného metóda je veľmi dobre spracovateľná do formy softvéru. Webové ale i mobilné aplikácie sú vhodné pri inej vizualizácii prostredí, ktoré deti poznajú v papierovej forme a práce v nich. V tejto časti sa pozrieme bližšie na niektoré z nich, aby sme mali lepšiu predstavu o tom, ako by naša aplikácia mala vyzerat', ktorým veciam by sme sa mali vyvarovať a ktorými sa práve naopak inšpirovať.

1.4.1. Úlohy z matematiky pre deti na základných školách, 2015-2019

Webová aplikácia Matika.in [3], ktorej autorom je Andrej Probst spracúva viacero prostredí Hejného metódy. Práca autorov príkladov sa však dá vnímať len ako náhľad do jednotlivých prostredí. Tútori Hejného metódy preto neuznávajú toto spracovanie ako jeden z vhodných prostriedkov na vyučovanie pomocou nej.

Prostredie Pavučín je spracované pre prvý až piaty ročník, avšak jediný typ úlohy, ktorý sa tu objavuje je úloha typu „doplniť čísla do prázdnych štvorčekov“.

Medzi pozitíva aplikácie patrí dodržiavanie časti metódy, ktorá hovorí o tom, že výsledok sa nemá nikdy prezrádzať. Ak dieťa úlohy vyrieši, klikne na tlačidlo „Mám hotovo!“, pokiaľ je vyplnený výsledok zlý, dieťa o tom informuje červený nápis, ktorý sa objaví a žiada o opravu jeho riešenia. Grafické spracovanie je pekné, šípky v grafe sú rôznofarebné, čo zaručene zaujme detského používateľa. Na vyplnenie čísla stačí nadísť nad

prázdne miesto myšou a môžeme zapísať náš výsledok. Aplikácia odmeňuje deti možnosťou vytlačenia diplomu ako odmeny pre dieťa.

Ako prvý problém, ktorý je takmer okamžite badateľný je nápis „Ako riešiť úlohu“, ktorý sa nachádza priamo pod úlohou. Hejného metóda sa snaží o to, aby dieťa prichádzalo na všetko samo bez návodu. Samozrejme, že pri práci s novým prostredím potrebujeme informáciu o tom, ako prostredie funguje. Konkrétne pri prostredí pavučiny stačí informácia o tom, že dieťa má doplniť čísla k šípkam a do krúžkov. Návod v tejto aplikácii však hovorí okrem toho aj to, že dieťa musí pričítať hodnotu šípky ku krúžku, z ktorého šípka vychádza a výsledok zapísať do krúžku, ku ktorému šípka smeruje. Týmto dieťa pripravuje o radosť z objavovania.

Rovnako je problém, že dieťa rieši len jeden typ pavučiny, v ktorom je gradácia úloh vyriešená zmenou farby a orientácie šípok, prípadne prácou s väčšími číslami, čo nepovažujeme za práve vhodný typ gradácie.



ÚLOHA Č. 3/5

MÁM HOTOVO! OBNOVIT!

Jak řešit úlohu?

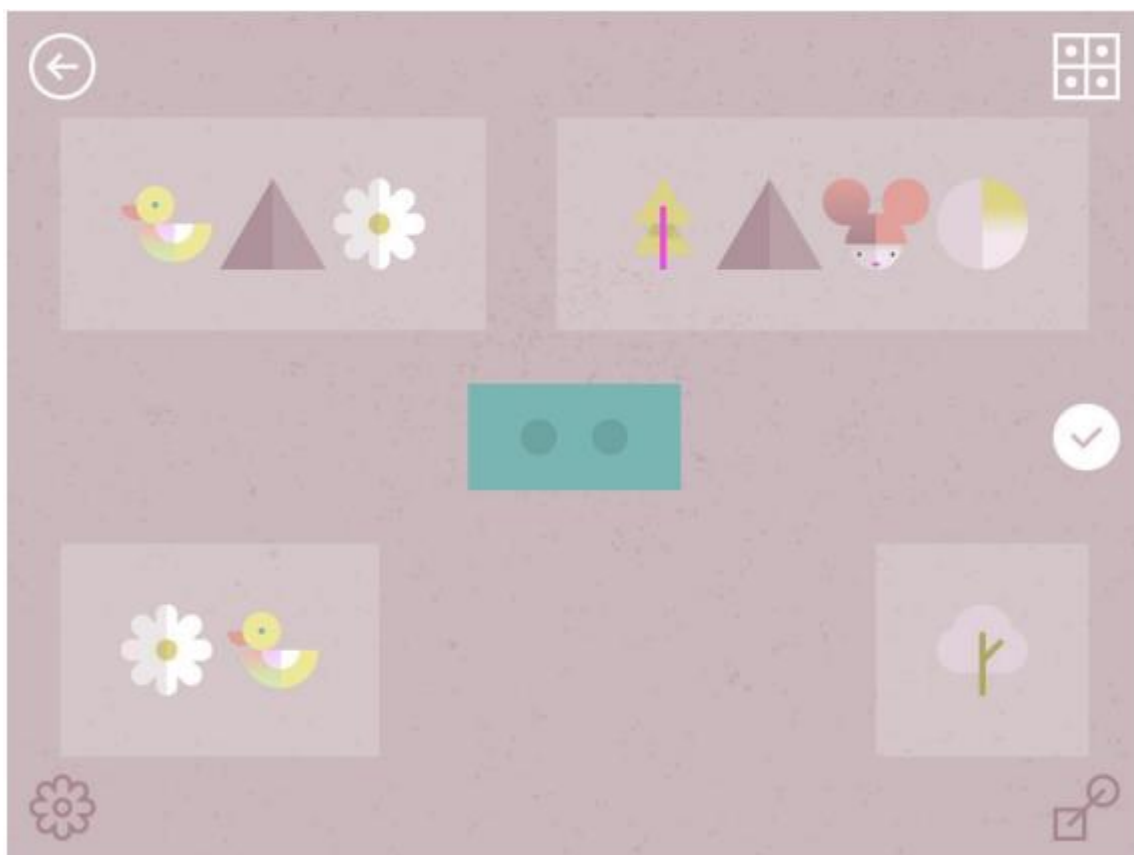
Napiš čísla do prázdných čtverečků v pavučině a k barevným šípkám pod pavučinou. Barevné šípky určují, kolik musíš přičíst k číslu ve čtverečku, odkud směřuje šípka. Výsledek pak zapiš do čtverečku, kde šípka končí.

Obrázok č.2: Úlohy z matematiky pre deti na základných školách

1.4.2. Dora is counting to 10 – learn math with fun

Hra [4] od slovenského vývojárskeho a grafického štúdia become mobile je určená pre deti vo veku od štyroch rokov. Je dostupná pre operačný systém iOS, teda zariadenia iPhone a iPad. Hra nepoužíva priamo Hejného metódu, je ňou iba inšpirovaná, pretože Hejného metóda sa zameriava na deti v školskom veku. Táto hra deti naučí počítat' do 10 a zároveň sa stretnú s jednoduchým sčítaním či odčítaním.

Veľkou výhodou aplikácie je, že sa zameriava na pochopenie čísel pomocou vizualizácie, presne tak, ako to je dieťaťu prirodzené. Napríklad v príklade, v ktorom je našou úlohou priradiť číslo dva k správnejmu počtu objektov, číslo je vyjadrené ako dve bodky a nie číslovkou. Zásady Hejného metódy môžeme vidieť napríklad v tom, že úlohy sú vytvorené intuitívne, bez toho, aby dieťa dostalo návod. Ale aj v tom, že aplikácia neprehráva správne riešenie. Aj keď je aplikácia prácou Slovákov, dabing je len v angličtine, no stále sa môže chváliť zvukovými efektami.



Obrázok č 3: Dora is counting to 10

1.4.3. Dora in a secret hole

Táto hra [9] je voľným pokračovaním vyššie spomenutej hry *Dora is counting to 10* od rovnakého autora - vývojárskeho a grafického štúdia become mobile. Rovnako ako prvá hra, i táto hra je inšpirovaná Hejného metódou. V hre je našou úlohou dostať sa z chodby von pomocou skákania po kamienkoch k východu. Používateľ sa na tomto princípe učí počítať s kladnými aj zápornými číslami do 20.

Na začiatku hry používateľ určuje smer a spôsob skákania hlavnej hrdinky Dory pomocou jednoduchých inštrukcií zadávaných prostredníctvom šípok v smere hore a dole. Dieťa si takto ani neuvedomuje, že na pozadí si takto buduje logiku počítania so zápornými číslami. S konkrétnymi číslami sa dieťa stretáva až v neskorších fázach hry, kde sa nimi nahradia šípkové inštrukcie.

Aplikácia dodržiava zásady Hejného metódy napríklad v intuitívnosti úloh, v tom, že aplikácia neprezeráda používateľovi riešenia. Dieťa taktiež získava v priebehu hry trofeje, ktoré menia hlavnej hrdinke farbu vlasov.



Obrázok č.4: Dora in a secret hole

1.4.4. Matemág – Edukačná hra pre deti

Matemág je edukačná hra pre deti vytvorená TechSophia s.r.o v spolupráci s Prof. RNDr. Milanom Hejným, CSc. Je určená pre žiakov prvého až tretieho ročníka. Matematika

je tu prezentovaná ako „matemágia“. Pekne graficky spracovaný príbeh hry je poprepletaný intuitívnymi úlohami a doplnený dobrým zvukovým spracovaním.

Príbeh hry sa zameriava na dve deti, ktoré cestujú za Matemágom. Našou úlohou je sprevádzať ich na tejto ceste a riešiť s nimi rébusy, na ktoré počas tejto cesty narazia. Hra je dabovaná a obsahuje interaktívne komiksy. Pri opakovanom spustení sa generátor úloh postará o to aby bola hra pre deti stále zábavná a atraktívna.

Hra je k dispozícii zdarma, no jej verzia pre školy je spoplatnená, rovnako ako i hry z tohto prostredia zamerané na konkrétne prostredia Hejného metódy.



Obrázok č.5: Matemág

1.4.5. Emil

Emil je metóda vyučovania informatiky na základných školách, ktorej autormi sú prof. RNDr. Ivan Kalaš, PhD. a jeho spolupracovníci. Okrem samotnej práce so softvérom pre počítače, či tablety taktiež pracuje s pracovnými zošitmi. Na prácu s touto metódou je nutné, aby učiteľ absolvoval školenie. Na tomto riešení môžeme vidieť, že hoci sa Hejného metóda zaoberá vyučovaním matematika, je možné využiť jej princípy aj inde.

Deti plnia úlohy spolu s Emilom, tie im často nepovedia, čo presne musia urobiť, ale skôr o čo sa majú pokúsiť. Jedno zo zadaní napríklad nabáda, aby pozbierali čo najviac

hrušiek, no na ďalšiu úlohu sa dostanú, aj bez nazbierania maximálneho počtu. Veľmi záleží na tom, či sú so svojim výsledkom spokojné. Tu ostáva priestor na motiváciu od svojich spolužiakov, ktorí dosiahli lepší výsledok.

Emil učí deti analyzovať a interaktívne riešiť problémy. Informatické myslenie je u detí rozvíjané hravou formou, pomocou ktorej sú motivované k väčšiemu záujmu.



Obrázok č.6: Emil

1.5. Predchádzajúce bakalárske práce na tému Hejného metódy

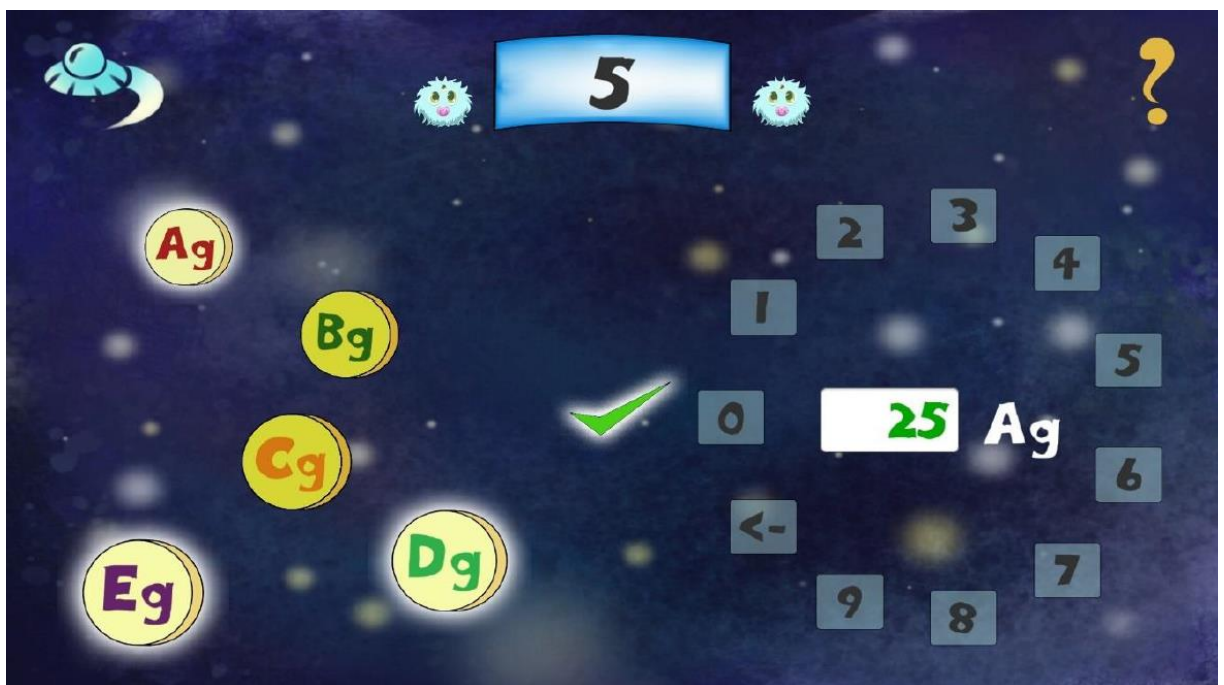
Hravá forma vyučovania sa v dnešnej dobe teší čoraz väčšej popularite. Práve preto existuje niekoľko bakalárskych prác, ktoré sa zaoberajú tvorbou edukačného softvéru. Neustále napredujúce vývojové prostredia, v ktorých je možné vytvoriť takéto hry, umožňujú každý rok lepšie spracovávať témy. Nahliadneme teda na dve práce, z ktorých jedna je vypracovaná v roku 2019 a druhá rok pred ňou.

1.5.1. Prostredie Biland

Hra Bilandia je výsledkom bakalárskej práce Júlie Gáblíkovej z roku 2019 [8]. Prostredie Biland, na ktoré je hra zameraná učí deti pracovať v dvojkovej sústave a v Hejného metóde je považované za jedno z ťažších prostredí.

Úloha používateľa sa v každej úrovni mení. V prvej úrovni premieňa bilandské sumy na A-groše a naopak. Následne sú zavedené základy sčítania a odčítania pomocou bilandských súm. Tretia úroveň je zameraná na prepis bilandských súm do jednotiek a núl. V poslednej úrovni pracuje používateľ už iba s číslami zapísanými v dvojkovej sústave, sčítava a odčítava ich, no s tým háčikom, že výsledok je už daný a úlohou je doplniť chýbajúce cifry v číslach.

Hra je veľmi pekne graficky spracovaná. Príšerky, ktoré sprevádzajú používateľa celou hrou sú zaujímavé, pre používateľa je pútavé ako sa príšerky postupne vyvíjajú. Jediná vec, ktorú by som na grafickom spracovaní zmenila, je ladenie do tmavých farieb. Pre žiakov základnej školy sú vhodnejšie pestré farby na udržanie pozornosti.



Obrázok č. 7: Bilandia

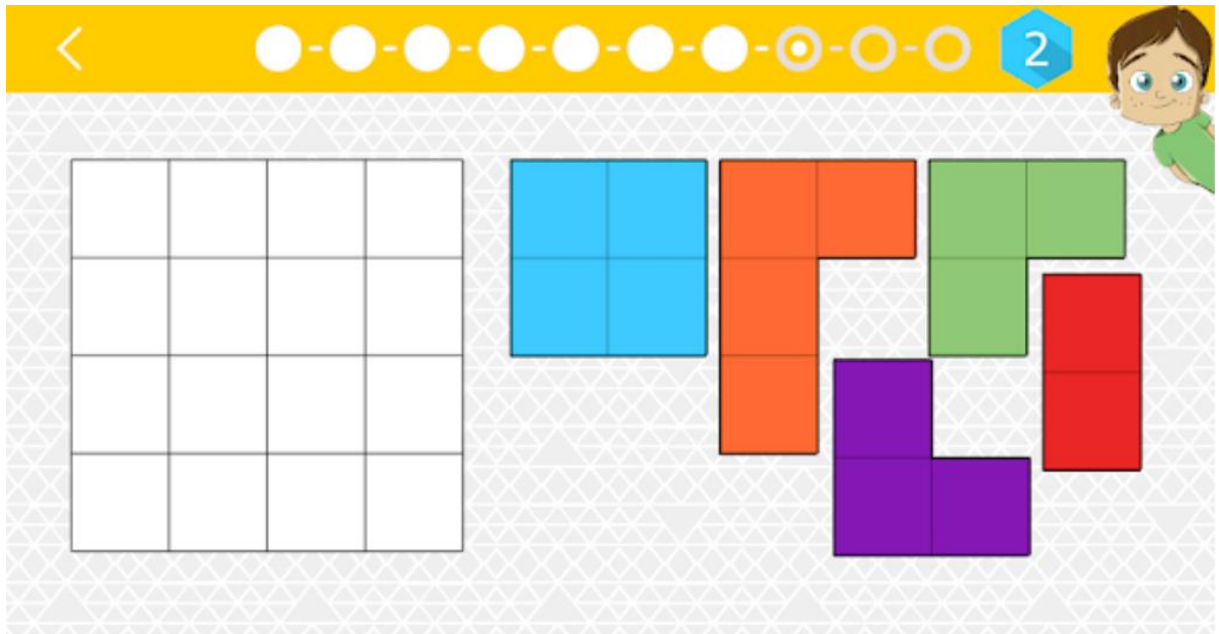
1.5.2. Prostredie parketovanie

Hra Parkety je výsledkom bakalárskej práce Andrey Spišákovej z roku 2018 [9]. Úloha užívateľa je poukladať parkety, útvary, ktoré sú poskladané zo štvorcov, do štvorcovej siete tak, aby ju celú vyplnili.

Táto aplikácia neobsahuje žiaden písaný text, namiesto toho je v aplikácii kreslený panáčik, ktorý je nahovorený a sprevádza používateľa úlohami. Okrem toho, po kliknutí na

panáčika, je možné pripomenúť si, čo je cieľom úlohy. Grafika je spracovaná veľmi prehľadne a farebne, čo je pútavé pre cieľovú skupinu používateľov.

Malým problémom je, že po odomknutí novej úrovne sa na lište zobrazí obrázok totožný s tým, ktorý nás z hlavného menu preniesie do úrovne. Túto funkcionality nemá pri kliknutí naň v spomínanej lište. Ak by tlačidlo malo takúto funkcionality aj na lište, prechod do ďalšej úrovne by bol jednoduchší.



Obrázok č.8: Parkety

1.6. Analýza technológií

Pri vytváraní akejkoľvek aplikácie je prvým krokom zvoliť prostredie, v ktorej budeme aplikáciu vyvíjať. Pre vytvorenie tejto aplikácie sme mali dve možnosti. Prvou bolo Android Studio, ktoré je vývojovou platformou pre zariadenia so systémom Android. Ako programovací jazyk využíva Javu s XML. Druhou možnosťou, pre ktorú sme sa nakoniec rozhodli, je prostredie Unity. Dôvodom pre toto rozhodnutie bolo, že Unity podporuje vývoj pre viacero operačných systémov, čiže aplikácia bude dostupnejšia.

1.6.1. Unity

Unity je pomerne rozšírený multiplatformový herný engine. Aj vďaka integrovanému vývojovému prostrediu (IDE) je ľahko pochopiteľný pre začiatočníkov a vhodný na použitie pre tento typ používateľa. Toto prostredie umožňuje vytvárať hry v 2D

a rovnako v 3D. Vďaka tomu, že Unity podporuje vývoj pre mobilné zariadenia s operačným systémom Android, iOS a Windows Phone, aplikácia bude omnoho dostupnejšia pre viacerých používateľov. Programovacie jazyky, ktoré Unity podporuje sú JavaScript a C#.

1.6.2. C#

C# je moderný, jednoduchý, objektovo-orientovaný jazyk od spoločnosti Microsoft. Je vhodný aj pre začiatočníkov. Ďalšou výhodou tohto jazyku je, že mnoho návodov a dokumentov na internete vysvetľuje použitie práve tohto jazyka v kombinácii s Unity. Tieto dôvody zavážili pri výbere C# nad JavaScriptom.

1.6.3. JetBrains Rider

Rider je multiplatformové vývojové prostredie od firmy JetBrains s integrovaným vývojovým prostredím (IDE). Rider podporuje mnohé jazyky okrem C#, v ktorom bude naša aplikácia naprogramovaná, podporuje napríklad jazyky F#, JavaScript, TypeScript. Vďaka podpore týchto jazykov umožňuje vytvárať konzolové aplikácie, knižnice, alebo hry v Unity. Prostredie uľahčuje vývojárovi prácu mnohými spôsobmi, napríklad dopĺňaním kódu, generovaním častí kódu, alebo veľkým množstvom refaktorizácií kódu. Súčasne obsahuje aj Unity editor.

2. Návrh

Pri procese vytvárania aplikácie, je dôležité dobre si premyslieť, ako má vyzerat' a fungovat'. Programovat' nezačneme bez toho, aby sme si najprv neutvorili predstavu o našej aplikácii. Pravdepodobne by sme veľmi skoro prišli na to, že pri programovaní musíme prerábať a mazať veľké časti už napísaného kódu. Komplikáciám sa vyhneme, vytvorením návrhu, ktorý si v tejto kapitole priblížime. Aby sme sa lepšie oboznámili s aplikáciou, postupne sa pozrieme na grafický dizajn a logiku jednotlivých úrovní hry.

2.1. Grafické rozhranie

Grafický dizajn hry je navrhnutý na šírku, objekty budú zobrazované na obrazovke prehľadnejšie. Zapnutá funkcia automatickej orientácie nezmení rozloženie obrazovky pri otočení na výšku.

2.2. Obrazovky výberu

Aplikácia má dve obrazovky slúžiace na výber spomedzi štyroch náročností úloh. Rozdelenie je nasledovné. Prvá obrazovka, slúži na výber levelu hry. Druhá obrazovka slúži na výber náročnosti úloh, ktoré chceme vytvárať.

2.2.1. Úvodná obrazovka

Úvodná obrazovka, inak povedané hlavné menu, sa zobrazí po zapnutí hry. Pozostáva z jednoduchého farebného pozadia s prechodom z tmavšej farby do svetlejšej. Vo vrchnej časti obrazovky je umiestnených niekoľko pavučín. Majú iba estetickú funkciu. Približne v strede obrazovky je sedem tlačidiel v tvare pavúkov a nadpis „PAVUČINY“.

V pravej spodnej časti obrazovky sa nachádzajú dve menšie tlačidlá. Prvé z nich má tvar pavúka s ozubeným kolieskom na chrbte. Toto tlačidlo nás prepne do malého menu, v ktorom je možné nastaviť hlasitosť zvukových efektov v hre a reštartovať ktorúkoľvek úroveň náročnosti s generovanými príkladmi, na prvú úlohu. Dôležitosť tejto funkcie si vysvetlíme neskôr, pri oboznámení sa s logikou jednotlivých úrovní. Ozubené koliesko intuitívne nabáda používateľa hľadať pod ním nastavenia. Druhé tlačidlo, umiestnené v rohu tejto obrazovky, je tlačidlo na vypnutie aplikácie. Vyzerá ako pavúk s celosvetovo známym symbolom pre zapnutie a vypnutie na chrbte.

Najväčšiu časť obrazovky zaberá päť takmer rovnakých tlačidiel v tvare pavúkov. Majú rovnaký dizajn ako ostatné tlačidlá na obrazovke. Prvé štyri z nich slúžia na zapnutie jednotlivých úrovní s generovanými príkladmi. Tieto tlačidlá majú tvar pavúka s číslami od

jedna po štyri na chrbte, čísla znamenajú náročnosť úloh. Piate tlačidlo má tvar pavúka so štetcom na chrbte. Zavedie nás na obrazovku výberu editora pre jednotlivé úrovne náročnosti. Týchto 5 tlačidiel sa líši veľkosťou. Pavúky, postupne s pribúdajúcou náročnosťou úloh, skrývajúci sa za nimi, rastú.

Po prvom spustení hry sú aktívne iba tlačidlá na vypnutie hry, nastavenia hlasitosti a reštartovania úrovni, prechod na obrazovku výberu úrovne editora a na prechod do prvej úrovne hry. Ostatné tlačidlá na prechod do úrovni hry sú neaktívne a odomknú sa až počas hry.



Obrázok č.9: Hlavné menu

2.2.2. Obrazovka výberu editora

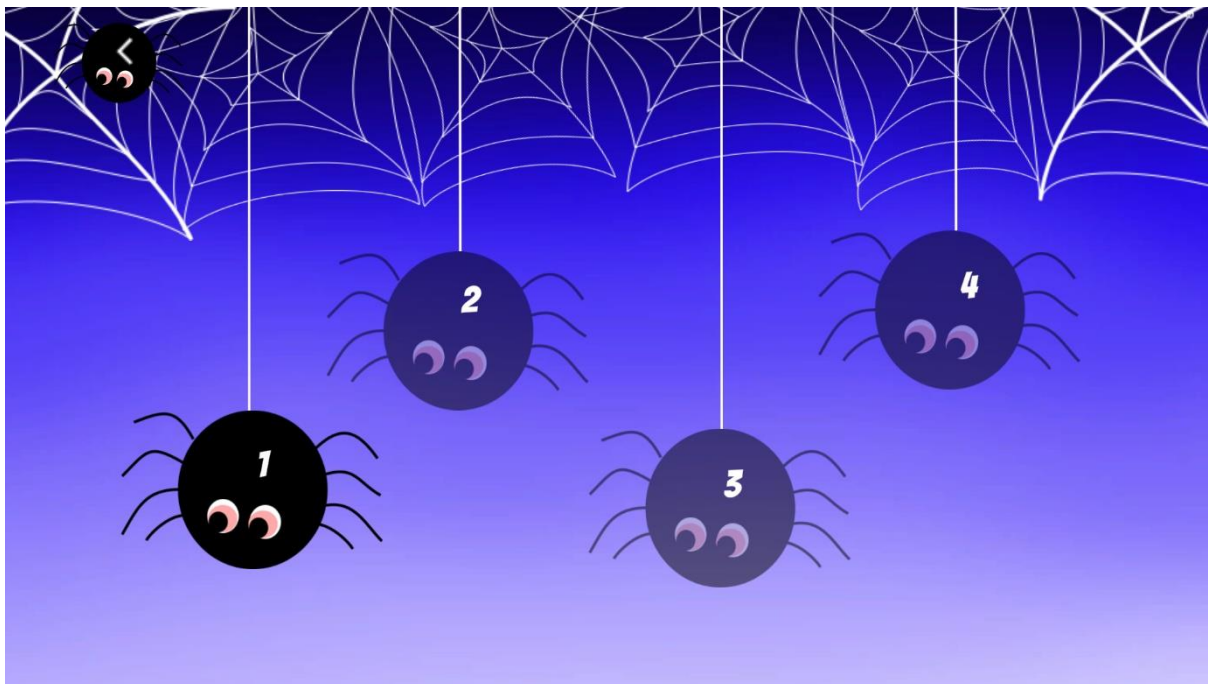
Táto obrazovka slúži na výber úlohy, ktorú chce používateľ vytvoriť. Dizajnom pripomína hlavné menu. Pozadie na tejto obrazovke je rovnaké ako v hlavnom menu, aby bolo hneď jasné, že aj táto obrazovka slúži na výber toho, čo zamýšľame robiť. Rovnako aj tu sú na vrchu obrazovky rozmiestnené pavučiny.

V ľavom hornom rohu je umiestnené tlačidlo v tvare pavúka so šípkou na chrbte. Toto tlačidlo slúži na návrat do hlavného menu.

Najväčšiu časť obrazovky opäť zaberajú tlačidlá v tvare pavúkov, tentokrát sú len štyri. Každé tlačidlo je označené číslom na pavúkovom chrbte a rovnako ako

v predchádzajúcom menu odlišuje náročnosť úlohy, ktorú chceme vytvoriť. Každé z týchto tlačidiel nás zavedie na obrazovku pre vytváranie úlohy danej náročnosti.

Ak otvoríme toto menu na začiatku hry, všetky tlačidlá budú neaktívne, pretože sme editor daného typu úlohy ešte neodomkli. O tom ako sa editor odomyká si viac povieme pri popise jednotlivých úrovní.



Obrázok č.10: Menu výberu editoru

2.3. Návrh úrovní

Prostredie pavučín je používané na výučbu v mnohých ročníkoch, no neobsahuje veľké množstvo typov úloh. Malá variácia úloh je dôvodom prečo sú si obrazovky pre každú úroveň veľmi podobné.

Aplikácia nedovoľuje používateľovi začať inou ako prvou úrovňou. Editor tejto úrovne sa odomkne zároveň s možnosťou riešiť ďalšiu úroveň náročnosti, po vyriešení deviatich príkladov v úrovni. Ďalšie úrovne s editormi sa odomykajú analogicky.

Úlohy sa v jednotlivých leveloch generujú. V levele sú zoskupené po štyroch úlohách s podobnou náročnosťou. Náročnosť úloh sa postupne zvyšuje. Tempo zvyšovania náročnosti úloh nemusí vyhovovať každému používateľovi, preto aplikácia umožňuje návrat k prvej úlohe v každej úrovni bez toho aby sa zablokovali iné funkcie, ktoré používateľ už odomkol. Tento návrat je možné urobiť v nastaveniach v hlavnom menu.

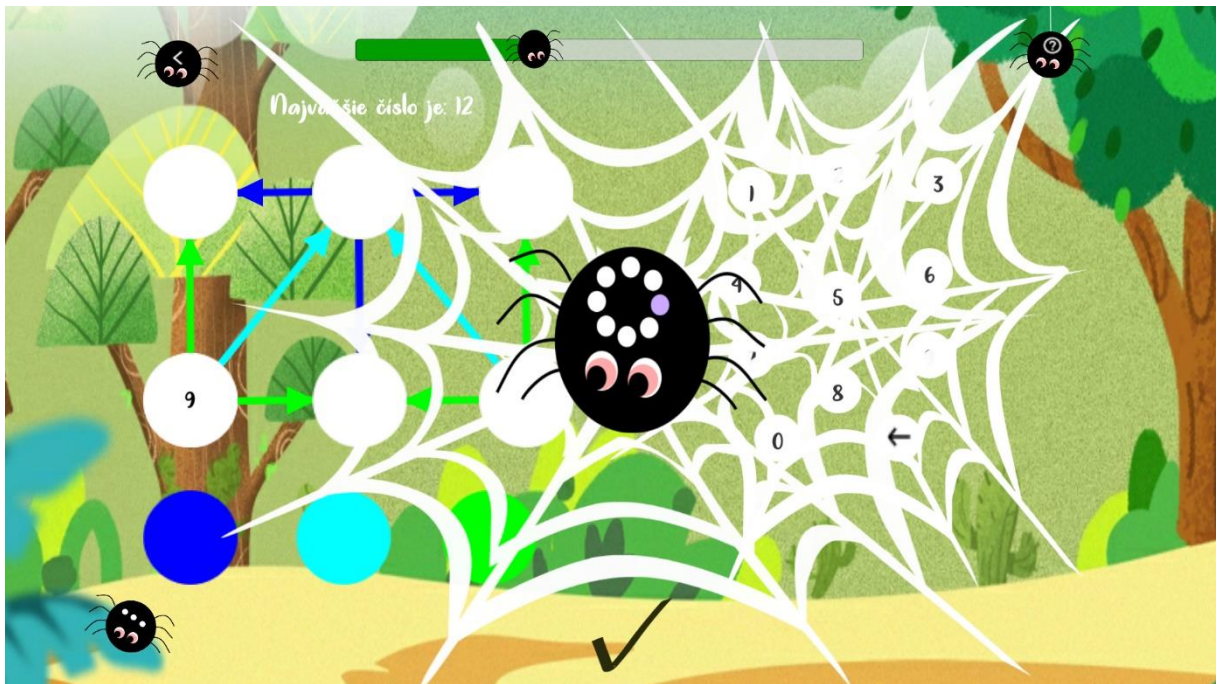
Na každej obrazovke, o ktorej si v tejto kapitole povieme sa nachádzajú objekty, ktoré sa nemenia pri zmene levelu. V ľavom hornom rohu sa nachádza tlačidlo v tvare pavúka so šípkou na chrbte, ktoré slúži na návrat do hlavného menu.

Na opačnej strane, v pravom hornom rohu sa nachádza tlačidlo v tvare pavúka s otáznikom na chrbte. Pomocou tohto tlačidla otvoríme obrázkový návod pre riešenie príkladov v danej úrovni.

Medzi týmito dvomi tlačidlami sa nachádza progres bar, ktorý nám hovorí o tom, koľko príkladov sme už v danej úrovni vyriešili.

Polovicu obrazovky zaberá klávesnica, ktorá vyzerá ako pavučina. Na pavučine sú pozachytávané číslice od nula do deväť a znamienko slúžiace na zmazanie číslice. Toto znamienko je dôležité v prípade, ak by sme sa v riešení pomýlili. Klávesnica funguje na princípe „Drag and Drop“. Ak používateľ potrebuje počas hry vytvoriť viacciferné číslo, je to možné urobiť tak, že vyplní najprv prvú číslicu a následne k nej potiahne druhú číslicu. Teda, ak by vytváral napríklad číslo 13, najprv do políčka potiahne číslicu 1 a následne k nej pridá číslicu 3. Ak by sa zmýlil, môže číslice mazať odzadu pomocou tlačidla, ktoré je na to určené.

Ďalšou a zároveň poslednou spoločnou črtou všetkých úrovní je potvrdenie riešenia. V spodnej časti obrazovky sa nachádza fajka, po ktorej stlačení sa úloha skontroluje. Ak je zadané riešenie nesprávne, v strede obrazovky sa objaví veľké preškrtnuté znamienko rovná sa, ktoré hovorí, že úloha nie je správne vyriešená. Ak je úloha správna, objaví sa animácia načítania, ktorá zaujme pozornosť dieťaťa kým sa načíta ďalšia úloha. Táto animácia sa zobrazí ako veľká pavučina cez celú obrazovku, v ktorej strede sedí pavúk, na ktorého chrbte sa krúti koliesko, ktoré symbolizuje načítavanie úlohy.



obrázok č.11 – načítanie príkladu

2.3.1. Prvá úroveň

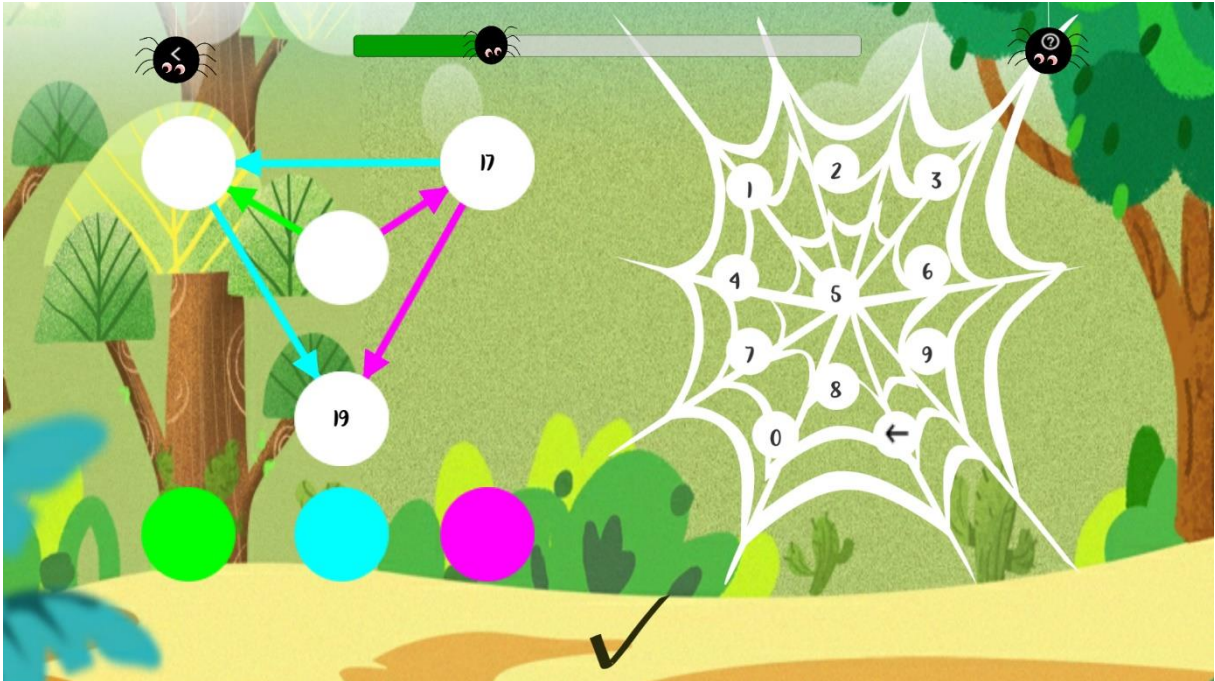
V prvej úrovni používateľ rieši pavučiny v tvare trojuholníka s tromi vrcholmi a štvrtým vrcholom uprostred. Znalosť ktorú potrebuje na vyriešenie je vzťah medzi jednotlivými vrcholmi, ktorý je naznačený orientovanými rôznofarebnými šípkami.

Na scéne, ktorú sme si vyššie popísali pribúda ku klávesnici približne rovnako veľký graf so štyrmi vrcholmi. Vrcholy grafu predstavujú vykreslené štyri biele krúžky. Hrany grafu sú vykreslené ako farebné šípky. Pod grafom sa nachádzajú krúžky vo farbách totožných s použitými farbami na vykreslenie jednotlivých hrán.

Úloha je pre používateľa v každom type úlohy rovnaká, musí zistiť hodnoty všetkých hrán a vrcholov. Hodnoty do jednotlivých vrcholov a krúžkov, ktoré vyjadrujú hodnoty hrán, používateľ zadáva ťahaním číslíc z klávesnice.

Náročnosť úloh sa stupňuje postupne po vyriešení štyroch približne rovnako náročných úloh. Najľahšie úlohy v tejto úrovni majú zobrazené dve hodnoty vrcholov grafu. Medzi týmito dvomi vrcholmi určite existuje hrana. Druhou možnosťou je, že používateľ vidí hodnotu jedného vrcholu a zároveň vidí i jednu hodnotu hrany. V náročnejších úlohách používateľ vidí zobrazené dva vrcholy v grafe, no tieto vrcholy nie sú priamo prepojené hranou. Vrcholy sú prepojené cez iný, nevyplnený vrchol dvoma hranami rovnakej hodnoty. Najnáročnejšie úlohy v tomto levele sa podobajú tým predchádzajúcim, no používateľ musí

okrem vrcholov a hodnôt hrán doplniť chýbajúcu hranu. Neznáma hrana je vykreslená ako šedá čiara bez určenia smeru šípkou.



obrázok č.12 – príklad prvej úrovne

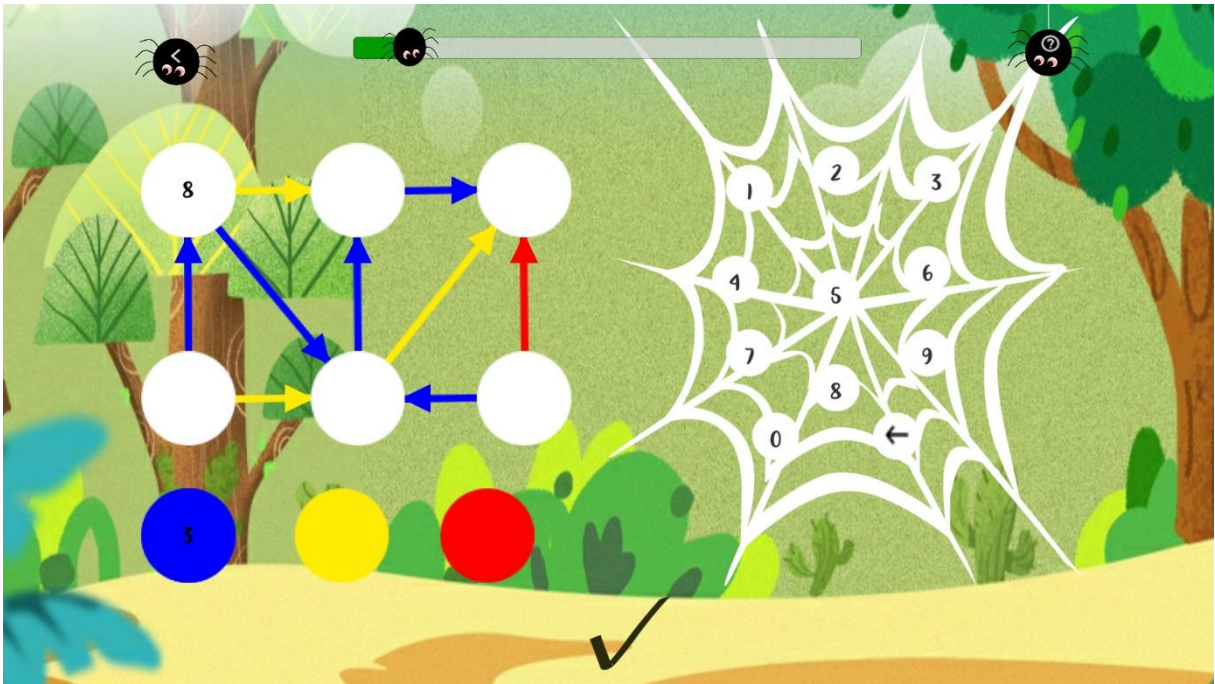
2.3.2. Druhá úroveň

V druhej úrovni sa používateľ popasuje s pavučinami, ktoré sú väčšie ako tej prvej. Tieto pavučiny majú šesť vrcholov. Rovnako, ako v predošlom leveli, potrebujeme na vyriešenie poznať vzťahy medzi vrcholmi, ktoré sú naznačené orientovanými šípkami.

Scéna je veľmi podobná tej z prvej úrovne, jediným rozdielom je graf, ktorý je v tejto úrovni väčší. Krúžky pod grafom sa nemenia, tie ostávajú rovnaké ako v predchádzajúcej úrovni.

Úlohou používateľa je opäť doplniť hodnoty vrcholov a hrán do grafu zobrazeného na obrazovke. Rovnako, ako v predošlej úrovni, to urobí ťahaním čísiel z klávesnice.

Opäť sa stretávame s gradáciou v skupinách po štyroch úlohách. Tieto úlohy sú veľmi podobné tým, s ktorými sa používateľ už stretol v predchádzajúcom leveli. Ako najjednoduchšie úlohy rieši také, ktoré majú zobrazenú hodnotu dvoch vrcholov a medzi týmito dvoma vrcholmi existuje hrana. Druhou alternatívou sú úlohy s jedným zobrazeným vrcholom a hodnotou jednej hrany, ktorá patrí tomuto vrcholu. V najťažších úlohách z tejto úrovne používateľ opäť do grafu doplní jednu neznámu hranu.



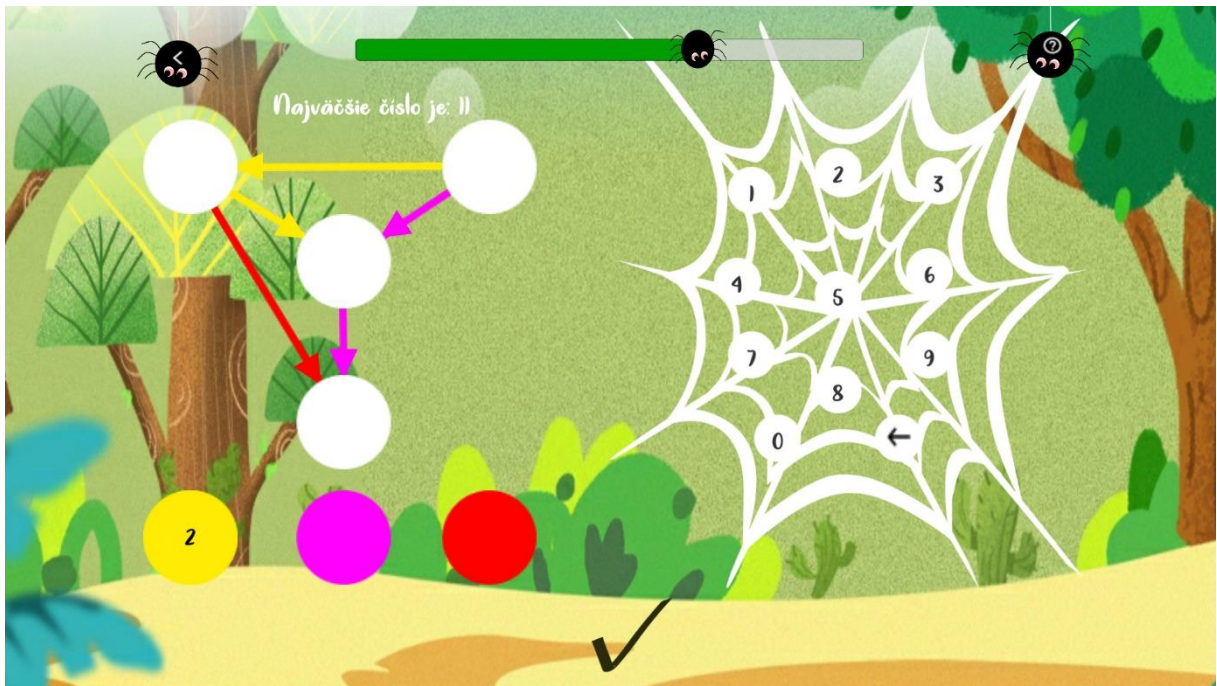
obrázok č.12 – príklad druhej úrovne

2.3.3. Tretia úroveň

V tretej úrovni sa používateľ opäť vracia k menším pavučinám so štyrmi vrcholmi, s ktorými sa už stretol v prvej úrovni. Tentokrát si však pri riešení úloh osvojí prácu s podmienkami.

V tejto úrovni je scéna takmer rovnaká ako v prvej. Jediná vec, ktorá odlišuje obrazovku tejto úrovne od prvej, je podmienka, ktorá sa zobrazuje nad grafom. Graf i krúžky symbolizujúce hodnoty hrán sa nemenia. Scéna môže byť jemne zmenená tým, že v ľavom dolnom rohu sa objaví tlačidlo pavúka s tromi bodkami na chrbte, ktorý nám otvorí prehľad už nájdených riešení, ak ich má úloha viacero.

Aj v tejto úrovni úlohy gradujú podobne ako v predchádzajúcich v skupinách po štyroch. Najjednoduchšie úlohy v tejto úrovni začínajú s vyplneným len jedným vrcholom, no okrem toho dostane používateľ i podmienku ohľadom grafu. Podmienka sa náhodne vyberie spomedzi - najväčšie číslo v grafe, najmenšie číslo v grafe. V náročnejších úlohách sa nám už v grafe nezobrazuje žiadne číslo, zobrazená je len jedna z vyššie spomenutých podmienok a hodnota niektorej z hrán v grafe. V najnáročnejších príkladoch v tretej úrovni dostávame pavučinu, teda graf, ktorý nemá vyplnený žiaden z vrcholov a rovnako nie je vyplnená žiadna z hodnôt hrán. Zobrazuje sa iba podmienka, ktorá môže byť v tomto prípade buď kombináciou vyššie spomenutých dvoch podmienok, teda najväčšie a najmenšie číslo v grafe alebo podmienka na súčet všetkých hodnôt vrcholov grafu.



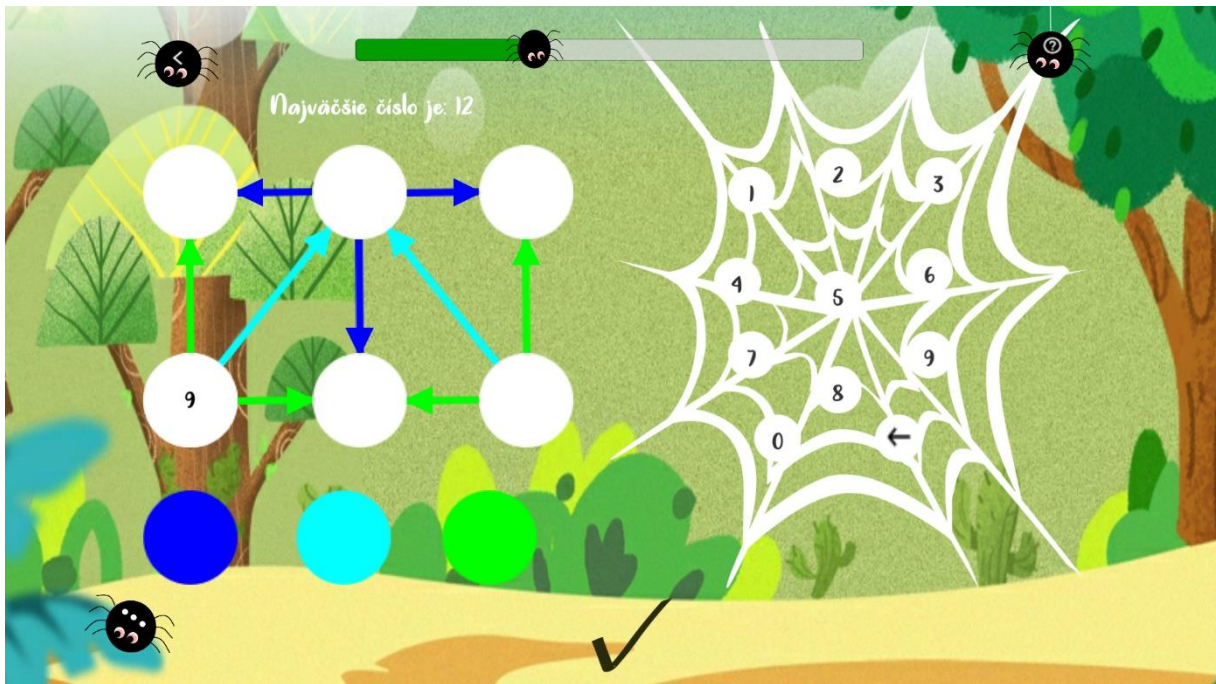
obrázok č.13 – príklad tretej úrovne

2.3.4. Štvrtá úroveň

V štvrtej, najťažšej úrovni sa používateľ popasuje opäť s väčšími pavučinami, ktoré majú 6 vrcholov. Na týchto pavučinách si osvojí prácu s podmienkami vo väčších grafoch.

Scéna sa ani v tejto úrovni príliš nemení. Pribúda opäť len podmienka ako v predchádzajúcej tretej úrovni. Rovnako aj na tejto obrazovke sa objavuje tlačidlo, ktoré otvorí prehľad už nájdených riešení.

V tejto poslednej úrovni opäť spôsob gradácie zachovávame. Príklady, ktoré sa riešia v tejto úrovni veľmi pripomínajú tie z predchádzajúcej úrovne, s tým rozdielom, že pavučiny, teda grafy sú väčšie. Pri najjednoduchších úlohách máme teda vyplnený len jeden vrchol v grafe a graf máme obmedzený nejakou podmienkou z – najväčšie číslo v grafe, najmenšie číslo v grafe. Zložitejšie úlohy, rovnako ako v predchádzajúcej úrovni, budú obmedzené niektorou z týchto dvoch spomenutých podmienok a hodnotou jednej hrany. Najťažšie úlohy nebudú mať opäť vyplnený žiaden vrchol, ani hodnotu hrany, budú len obmedzené, buď kombináciou dvoch vyššie spomenutých podmienok, alebo podmienkou na súčet hodnôt vo všetkých vrcholoch pavučiny.



obrázok č.13 – príklad štvrtej úrovne

2.4. Návrh editorov

Vzhľadom na fakt, že úlohy v prostredí pavučín sú si veľmi podobné i editory na vytváranie vlastných úloh majú veľa spoločných črt. V ľavej hornej časti sa nachádza tlačidlo s bielou šípkou, ktoré používateľa vráti na obrazovku výberu vytvárania úlohy. Každý editor obsahuje klávesnicu rovnakú ako v každej jednej úrovni. Rovnako veľkú časť obrazovky zaberá graf, ktorý má rovnaký tvar ako v príslušnej generovanej úrovni. Krúžky s hodnotami hrán majú po spustení generátora bielu farbu. Ich farbu je možné meniť ťuknutím. Grafy v generátoroch rovnako ako krúžky s hodnotami hrán majú všetky hrany na začiatku biele, nastaviť ich farbu je možné tiež pomocou ťukania. Posledným spoločným grafickým prvkom v editoroch je zelené tlačidlo v spodnej časti editora s nápisom „HOTOVO“.

V editoroch pre druhú a tretiu úroveň pribúda možnosť zvolenia podmienky použitím bielych šípok a doplnením čísla do bieleho krúžku pri podmienke.

Editor, ako súčasť aplikácie, je navrhnutý pre zdatnejších používateľov, aby si mohli vytvoriť vlastné zadanie úloh. Rovnako môže byť užitočný i pre vyučujúceho, ktorý pomocou editora môže vytvoriť úlohu pre svojich žiakov a zaradiť ju do procesu vyučovania. Učiteľ môže do aplikácie týmto spôsobom preniesť príklad z učebnice.

Editor je navrhnutý tak, aby pomáhal precvičovať matematické schopnosti používateľa pri vytváraní úloh, pri čom každá úloha musí mať aspoň jedno riešenie. Ak

používateľ vytvorí úlohu bez riešenie, alebo takú, ktorá má viac riešení, editor mu nedovolí spustiť úlohu. K editoru sa používateľ dostane iba v prípade, že z úrovne s danou náročnosťou vyrieši deväť príkladov. Toto obmedzenie je použité z dôvodu, aby mal používateľ lepšiu predstavu o tom, aký princíp použiť na riešenie úloh.



obrázok č.13 – príklad štvrtej úrovne

3. Implementácia

Na úspešné dokončenie projektu tak, aby zaujal cieľové publikum a zároveň bol funkčný, nestačí iba dobrý návrh. Rozdelenie práce na menšie úlohy pri implementácii je rovnako dôležité ako vytvorenie návrhu aplikácie. V tejto kapitole si preto postupne priblížime štruktúru aplikácie, implementáciu jednotlivých úrovní a editorov, a taktiež ovládanie našej aplikácie.

V tejto kapitole sú pridané ukážky zdrojového kódu, ktoré nám pomôžu lepšie si vysvetliť postup pri vytváraní aplikácie.

Aplikácia je vyvíjaná v Unity, multiplatformovom hernom engine. Aplikácia sa taktiež prispôsobuje rozmerom rôznych tabletov, ale i mobilných zariadení, teda je responzívna. Vďaka tomu, že Unity podporuje vývoj pre viacero operačných systémov, aplikácia bude dostupná v prvom rade pre zariadenia s operačným systémom Android, no takisto aj pre tie s operačným systémom iOS.

3.1. Štruktúra aplikácie

Projekt v engine Unity obsahuje zložky Assets, v nich sa nachádzajú priečinky, ktorých obsah tvoria stavebné prvky na vyskladanie používateľského rozhrania. Týmito prvkami sú scény, obrázky, animácie, zvuky, písma a najdôležitejšie z nich scripty.

Aplikácia vytváraná v Unity sa skladá zo scén. Týchto scien môže byť viacero a používateľ sa medzi nimi v priebehu hry presúva. Naša aplikácia má 3 scény. Každá zo scén má vlastné objekty a scripty. V našej aplikácii je hlavnou scénou *menu*, tá sa zobrazí ako prvá po spustení aplikácie. Na prepínanie sa medzi scénami používame triedu *SceneManager*, ktorá obsahuje viacero metód. Naša aplikácia používa metódu *LoadScene(string sceneName)*.

```
public void Playlevel1()
{
    PlayerPrefs.SetInt("levelClicked",1);
    PlayerPrefs.SetInt("editor",0);
    SceneManager.LoadScene("level1");
}
```

obrázok č.14 – ukážka kódu

Keďže každá scéna má svoje vlastné meno. Po stlačení tlačidla v hlavnom menu sa zavolá metóda, ktorá používateľa prepne do scény levelu, ktorý používateľ zvolil.

Základným objektom na každej scéne v prostredí Unity je herný objekt, slúži na reprezentáciu znakov, scenérií, či špeciálnych efektov. Aby sme z herného objektu vytvorili nami požadovanú vec, musíme mu priradiť špeciálne vlastnosti a komponenty, ktoré určia jeho vzhľad a správanie. V našej aplikácii sú napríklad hernými objektami krúžky na dopisovanie čísel, hrany grafu, či tlačidlá. Napríklad komponentom, ktorý obsahujú všetky tlačidlá, je komponent `Button`, ktorý zabezpečí, že po kliknutí na toto tlačidlo, sa vykoná akcia, ktorú mu priradíme. Herné objekty si môžeme sami vystavať pomocou kombinovania rôznych komponentov.

3.2. Implementácia úrovni

Ako sme už vyššie spomenuli, každá scéna v editore Unity obsahuje niekoľko objektov. Súčasťou každej scény je objekt `Canvas`, ktorý má priradený `CanvasScaler`. Tento `CanvasScaler` zabezpečuje čiastočnú responzivnosť aplikácie. Okrem objektu `Canvas` obsahuje každá scéna `EventSystem` zodpovedný za spracovanie udalostí.

Objekty sa na plochu vykresľujú v poradí, v akom sú zoradené v hierarchii v editore. Všetky objekty, ako napríklad tlačidlá, texty, klávesnica, progres bar, sú v scéne deťmi objektu `Panel`. Takýchto objektov sme použili pri vytváraní scény viaceré, aby sme logicky oddelili jednotlivé veci zobrazené na obrazovke. Vlastný panel má napríklad klávesnica, ktorý má vnorené objekty, slúžia ako tlačidlá klávesnice. Objekty, ktoré sa počas hry na obrazovke nemenia, sú tri tlačidlá. Dve sú v tvare pavúkov a nachádzajú sa v hornej časti obrazovky. V ľavom rohu slúži tlačidlo na návrat do hlavného menu, tlačidlo v pravom rohu otvorí pomôcku na riešenie úloh. Posledným tlačidlom je tlačidlo v tvare „fajky“, ktoré overí správnosť vyriešenej úlohy. Na sledovanie počtu už vyriešených príkladov, je použitý objekt `Slider`, ktorý poskytuje samotný editor Unity. V tomto objekte je možné jednoducho nastaviť jeho minimálnu a maximálnu hodnotu, aj to, že hodnota má byť iba celé číslo. Grafický dizajn tohto objektu je veľmi jednoducho prispôsobiteľný, v našom prípade bežec vyzerá ako pavúk a „prejdená časť“ má zelenú farbu.

V skripte pre ovládanie scény úrovne využívame metódu `Start()`, ktorú má v ponuke samotný editor Unity. V tejto metóde inicializujeme všetky objekty, ktoré sa majú na obrazovke zobraziť od zobrazenia obrazovky. Tie objekty, ktoré používateľ vidieť nemá, sa v tejto metóde skryjú. Táto metóda tiež zodpovedá za zobrazenie prvej úlohy po prepnutí na scénu konkrétnej úrovne.

3.3. Implementácia editora

Editor úloh má jednoduchý grafický dizajn s jednofarebným pozadím. V scéne pre editor sú objekty, ktoré sa nachádzajú zároveň aj v scéne pre hranie úrovne. Takýmito objektami je klávesnica, panely, či tlačidlo na návrat na obrazovku výberu. Výrazné je na scéne zelené tlačidlo „HOTOVO“, ktoré po vytvorení zadania, úlohu skontroluje. Ak má vytvorená úloha riešenie, užívateľ prejde na obrazovku riešenia úlohy. Odtiaľ sa po stlačení zelenej fajky dostane naspäť do editora.

3.4. Ovládanie aplikácie

Na ovládanie aplikácie sme využili systém *EventSystem*, ktorý sa na každej scéne nachádza iba raz. Tento systém zodpovedá za spracovanie udalostí na scéne. Vďaka tomuto systému je možné zistiť, s ktorým objektom na scéne používateľ interagoval. Túto vlastnosť sme využili pri tvorení klávesnice, ktorá funguje na potiahnutie a pustenie, a zároveň aj na klikanie na tlačidlá a hrany grafu.

Na vytvorenie klávesnice ovládanej pomocou potiahnutia a pustení – „Drag and Drop“ používa aplikácia scripty *DragHandler* a *Drop*. Na menenie hrán grafu je použitý script *ArrowChanger*. Na vytvorenie týchto scriptov je použitá knižnica *UnityEngine.EventSystems*.

Na implementáciu triedy *DragHandler* sú použité nasledovné rozhrania: *IBeginDragHandler*, *IEndDragHandler*, *IDragHandler*. Pri kliknutí na objekt, ktorý má priradený tento script sa volá metóda *OnBeginDrag()*, ktorá je zdedená z rozhrania *IBeginDragHandler*. Metóda si v našom prípade zapamätá objekt do globálnej premennej, rodičovský objekt a začiatočnú pozíciu tohto zakliknutého objektu. Pri pustení objektu sa volá metóda *OnEndDrag()*, zdedená z *IEndDragHandler*, ktorá nastaví globálnu premennú, v ktorej bol uložený objekt na hodnotu null. Počas ťahania objektu po obrazovke sa volá metóda *OnDrag*, zdedená z *IDragHandler*. Táto metóda prepisuje pozíciu zvoleného objektu na súradnice získané z pozície kurzora na obrazovke, v prípade interakcie na tablete alebo mobilnom telefóne je kurzorom náš prst.

Objekty do ktorých je žiadúce doplniť čísla na vyriešenie úlohy v aplikácií majú priradený script *Drop*. Tento obsahuje jednu metódu *OnDrop*, zdedenú z rozhrania *IDropHandler*. Po pustení objektu metóda zistí, či miesto, na ktoré chceme vložiť objekt, už neobsahuje iný objekt. V prípade, že neobsahuje iný objekt, nastaví objektu predstavujúcemu voľné miesto ako dieťa ťahaný objekt a na štartovacej pozícii sa vytvorí

kópia ťahaného objektu. Ak už ale obsahuje nejaký objekt ako svoje dieťa, overí sa, či je text dvojčiferné číslo. Ak text nie je dvojčiferné číslo, pridá k nemu ťahanú číslicu. Ak je textom dvojčiferné číslo, nič sa k nemu nepridá. V oboch prípadoch sa ťahaný objekt vráti na svoju štartovaciu pozíciu.

```
public class DragHandler : MonoBehaviour, IBeginDragHandler, IDragHandler, IEndDragHandler
{
    public static GameObject itemBeingDragged;
    private Vector3 startpos;
    private Transform startParent;

    public void OnBeginDrag(PointerEventData eventData) {
        itemBeingDragged = gameObject;
        startpos = transform.position;
        startParent = transform.parent;
        GetComponent<CanvasGroup>().blocksRaycasts = false;
    }

    public void OnDrag(PointerEventData eventData)
    {
        transform.position = Input.mousePosition;
    }

    public void OnEndDrag(PointerEventData eventData)
    {
        if (transform.parent == startParent)
        {
            transform.position = startpos;
        }
        itemBeingDragged = null;
        GetComponent<CanvasGroup>().blocksRaycasts = true;
    }
}
```

obrázok č.15: ukážka kódu

```
public void OnDrop(PointerEventData eventData)
{
    if (!Item) {
        GameObject itemBeingDragged = DragHandler.itemBeingDragged;
        if (itemBeingDragged.transform.parent.CompareTag("graf") && transform.childCount != 1) {
            if ("b" != itemBeingDragged.transform.GetChild(0).transform.GetComponentInChildren<Text>().text.ToString())
            {
                GameObject newGameObject = Instantiate(itemBeingDragged, itemBeingDragged.transform.parent.position, Quaternion.identity);
                itemBeingDragged.GetComponent<Image>().color = transform.GetComponent<Image>().color;
                newGameObject.transform.SetParent(itemBeingDragged.transform.parent, worldPositionStays: false);
                newGameObject.GetComponent<CanvasGroup>().blocksRaycasts = true;
            }
        }
        if (transform.childCount != 1) {
            if ("b" != itemBeingDragged.transform.GetChild(0).transform.GetComponentInChildren<Text>().text.ToString())
            {
                itemBeingDragged.transform.SetParent(transform);
                itemBeingDragged.GetComponent<Image>().color = transform.GetComponent<Image>().color;
            }
        }
        else
        {
            Text textFromParent = transform.GetChild(0).transform.GetChild(0).transform.GetComponentInChildren<Text>();
            Text textFromItemBeingDragged = itemBeingDragged.transform.GetChild(0).transform.GetComponentInChildren<Text>();
            if (textFromParent.text.Length == 1 && textFromItemBeingDragged.text.ToString() != "b")
            {
                textFromParent.text += textFromItemBeingDragged.text;
            }
            if (textFromParent.text.Length >= 1 && textFromItemBeingDragged.text.ToString() == "b")
            {
                textFromParent.text = textFromParent.text.Substring( startIndex: 0, length: textFromParent.text.Length-1);
                if (textFromParent.text.Length == 0)
                {
                    Destroy(transform.GetChild(0).gameObject);
                }
            }
        }
    }
}
```

obrázok č.16: ukážka kódu

Ďalším zaujímavým prvkom ovládania je menenie farby a orientácie hrán v grafe. Šedá hrana bez orientácie má priradený script *ArrowChanger*, ktorý obsahuje metódu *OnPointerClick*, zdedenú z triedy *IPointerClickHandler*. Na začiatku metóda získa počet kliknutí, štruktúru grafu a zistí farbu všetkých hrán v grafe za pomoci farebných krúžkov. Tieto farby si uloží do dátovej štruktúry *List* s názvom *colors*, s ktorou bude neskôr pracovať. Metóda nájde hranu, na ktorú sme klikli v štruktúre grafu. Funkcia odstráni objekt predstavujúci šípku zo scény. V štruktúre grafu sa podľa počtu kliknutí zmení parameter *invisibleColor* na ďalšiu farbu z vyššie spomenutej dátovej štruktúry *List* s názvom *colors*. Ak už sme prešli všetkými farbami v tejto štruktúre, je nutné zmeniť orientáciu hrany, to sa vykoná nastavením parametra *lastAngle* v štruktúre grafu. Nakoniec sa zavolá funkcia, ktorá zabezpečí vykreslenie hrany grafu nanovo.

```
public void OnPointerClick(PointerEventData eventData)
{
    int poc = make.poc;
    Graph g = make.newGraph();
    List<Color> colors = new List<Color>();
    foreach (var dot in g.dots)
    {
        if (dot.color != Color.gray)
        {
            colors.Add(dot.color);
        }
    }
    if (PlayerPrefs.GetInt( key: "editor")==1)
    {
        colors.Add(Color.gray);
        colors.Add(Color.white);
    }

    foreach (var edge in g.edges) {
        if (!edge.visible && ((edge.@from.name + "-" + edge.to.name) == gameObject.name || (edge.to.name + "-" + edge.@from.name) == gameObject.name )) {
            Destroy(gameObject);
            if (poc % (colors.Count * 2) < colors.Count) {
                if (poc % (colors.Count * 2) == 0)
                {
                    edge.invisibleAngle += 180;
                    edge.lastAngle = !edge.lastAngle;
                }
                edge.invisibleColor = colors[poc % (colors.Count * 2)];
                make.poc++;
            }
            else {
                if (poc % (colors.Count * 2) == colors.Count)
                {
                    edge.invisibleAngle += 180;
                    edge.lastAngle = !edge.lastAngle;
                }
                edge.invisibleColor = colors[(poc % (colors.Count * 2)) - colors.Count];
                make.poc++;
            }
        }
        if (edge.invisibleColor != Color.gray || edge.invisibleColor != Color.white) {
            Destroy( @obj: GameObject.Find(edge.@from.name + "-" + edge.to.name + "head"));
        }
        g.CreateArrow(edge, 30f, false);
    }
}
```

obrázok č.16: ukážka kódu

3.5. Generovanie a kontrola úloh

Prvotným zdrojom informácií a úloh pre dieťa je učebnica a pracovný zošit, s ktorým pracuje na vyučovaní. Ak dieťa potrebuje viac cviku pri riešení úloh, alebo ho riešenie úloh baví, učebnica s pracovným zošitom sú obmedzené počtom strán a úloh. Oproti týmto

zdrojom, ktoré majú obmedzený počet úloh, aplikácia zabezpečuje väčšie množstvo a variáciu úloh. Toto je zabezpečené prostredníctvom vhodnej práce s náhodnými číslami pri generovaní úloh, ktoré zabezpečuje generátor.

Za generovanie úloh v aplikácií zodpovedá jeden spoločný generátor. Tento generátor generuje úlohy podľa aktuálnej úrovne hry a počtu vyriešených úloh v používateľom zvolenej úrovni, tak ako sme si už priblížili vyššie v návrhu aplikácie.

Keďže generovanie náhodným grafov nie je triviálna úloha a ak by sme ich generovali s tromi náhodnými číslami, grafy by nevychádzali pekne pospájané. Na príkladoch z učebníc, či pracovných zošitov, si môžeme všimnúť, že hodnoty hrán spolu súvisia tak, aby sa graf jednoduchšie vytváral. Napríklad, ak má graf hrany s hodnotami 1, 2 a 3, je jednoduché s nimi pracovať, pretože väčšie čísla sa ľahko skladajú za pomoci tých menších. Ak by mali hrany hodnoty 3, 5 a 9, väčšie hodnoty nie je možné vyskladať za pomoci tých menších. Pri generovaní grafu nám pomáha práve tento princíp. Preto si vygenerujeme najprv číslo od 1 do 3 a následne jeho dvojnásobok a trojnásobok. Potom si pri grafoch so štyrmi vrcholmi vyberieme náhodný vrchol. Priradíme mu hodnotu, následne ho prepojíme náhodnou hranou s náhodnou orientáciou so všetkými jeho susednými vrcholmi. Pri grafoch so šiestimi vrcholmi toto vykonávame na vrchole v strede, pretože jedine tie susedia v grafe so všetkými vrcholmi. Takýmto spôsobom nám vzniknú hodnoty v každom vrchole. Prejdeme všetkými vrcholmi a pospájame ich s ich susedmi možnou hranou. Z takto vytvoreného grafu si podľa typu úlohy vyberieme vrcholy, ktoré používateľovi zobrazíme na obrazovke, poprípade hranu, ktorú používateľovi neukážeme. Po všetkých týchto krokoch graf skontrolujeme tak, akoby sme videli iba to, čo vidí používateľ a zistíme počet riešení. Ak má graf vhodný počet riešení, zobrazíme ho pre používateľa, aby ho mohol riešiť.

Používateľom zadané riešenie následne porovnáваме s riešeniami, ktoré sme našli.

Zdroje

[1] Virtual Lab: Premeny triedy [online]. [cit. 5.1.2020]. Dostupné online na internete:

<http://www.virtuallab.sk/claroline/claroline/backends/download.php?url=L1ByZW1lbnFfdHJpZWR5XzZlucGRm&cidReset=true&cidReq=M_DT>

[2] H-mat: Co je to „Hejného metoda“? [online]. [cit. 5.1.2020]. Dostupné online na internete:

<<https://www.h-mat.cz/hejneho-metoda>>

[3] Úlohy z matematiky pre deti na základných školách, 2015 – 2018 [online]. [cit. 5.1.2020]. Dostupné online na internete:

<<https://www.matika.in/sk/>>

[4] Dora is counting to 10 [online]. [cit. 5.1.2020]. Dostupné online na internete:

< <http://www.become.sk/portfolio/dora-is-counting-to-10-matematicka-hra-pre-deti/>>

[5] Matemág – Edukačná hra pre deti [online]. [cit. 5.1.2020]. Dostupné online na internete:

< <https://www.matemag.cz/>>

[6] Emil [online]. [cit. 5.1.2020]. Dostupné online na internete:

< <https://www.robotemil.com/sk/>>

[7] Júlia Gáblíková: Softvérová podpora vyučovania matematiky Hejného metódou – prostredie Biland, FMFI UK Bratislava, 2019

[8] Andrea Spišáková: Softvérová podpora vyučovania matematiky Hejného metódou – prostredie Parketovanie, FMFI UK Bratislava, 2018

[9] Dora in a secret hole [online]. [cit. 14.5.2020]. Dostupné online na internete:

< <http://www.become.sk/portfolio/edu-hra-dora-secret-hole/>>