

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

INTERAKTÍVNA WEBOVÁ HRA NA PODPORU DETSKEJ
LOGOPÉDIE
Diplomová práca

2025

Bc. Juraj Kováč

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

INTERAKTÍVNA WEBOVÁ HRA NA PODPORU DETSKEJ
LOGOPÉDIE
Diplomová práca

Študijný program: Počítačové siete
Študijný odbor: Informatika
Školiace pracovisko: Katedra elektroniky a multimediálnych telekomunikácií (KEMT)
Školiteľ: doc. Ing. Stanislav Ondáš, PhD.

2025 Košice

Bc. Juraj Kováč

Abstrakt v SJ

Predmetom diplomovej práce je tvorba webovej platformy, ktorá umožňuje gamifikovať logopedické cvičenia. Práca pozostáva z teoretickej časti, kde sa rieši problematika logopédie, logopedických porúch, cvičení a následne vysvetľuje pojem gamifikácia. V praktickej časti na začiatok prebieha vytvorenie návrhu a dizajnu jednotlivých podstránok. Následne prebieha realizácia samotnej platformy cez technológie ako HTML, CSS, JavaScript, PHP a MySQL. Rozpoznávanie reči je vykonávané pomocou Web Speech API, ktorá vracia rozpoznanú hodnotu v textovej podobe. Platforma je vytvorená na dynamické vytváranie hier podľa zvolených úloh, obrázkov a štruktúry mapy. Samotná hra funguje na princípe pohybu po hernom poli pomocou hernej kocky alebo správnej odpovede, kde na každom hernom poli sa vykonáva iný typ úlohy. Platforma obsahuje demo verziu hry, ktorá slúži ako demonštrácia funkcionalít platformy.

Kľúčové slova v SJ

logopédia, hra, gamifikácia, webová platforma, Phaser 3, Web Speech API, klient-server

Abstrakt v AJ

The subject of the master's thesis is the development of a web platform that enables the gamification of speech therapy exercises. The thesis consists of a theoretical part, which addresses the issues of speech therapy, speech disorders, exercises, and subsequently explains the concept of gamification. In the practical part, the process begins with the design and layout of individual subpages. Then, the actual implementation of the platform takes place using technologies such as HTML, CSS, JavaScript, PHP, and MySQL. Speech recognition is performed using the Web Speech API, which returns the recognized value in text form. The platform is designed for dynamically creating games based on selected tasks, images, and map structures. The game itself operates on the principle of moving across the game board using either a dice roll or a correct answer, with each game field presenting a different type of task. The platform includes a demo version of the game, which serves as a demonstration of the platform's functionalities.

Kľúčové slova v AJ

speech therapy, game, gamification, web platform, Phaser 3, Web Speech API, client-server

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY
Katedra elektroniky a multimediálnych telekomunikácií

ZADANIE DIPLOMOVEJ PRÁCE

Študijný odbor: **Informatika**
Študijný program: **Počítačové siete**

Názov práce:

Interaktívna webová hra na podporu detskej logopédie
An interactive web game to support children's speech therapy

Študent: **Bc. Juraj Kováč**
Školiteľ: **doc. Ing. Stanislav Ondáš, PhD.**
Školiace pracovisko: **Katedra elektroniky a multimediálnych telekomunikácií**
Konzultant práce:
Pracovisko konzultanta:

Pokyny na vypracovanie diplomovej práce:

1. Naštudovať oblasť vytvárania webových aplikácií a oblasť aplikácií pre detskú logopédiu.
2. Navrhnuť interaktívnu webovú hru pre logopedickú terapiu.
3. Implementovať interaktívnu webovú hru pre logopedickú terapiu.
4. Vypracovať záverečnú prácu opisujúcu analytickú a praktickú časť podľa pokynov vedúceho práce.

Jazyk, v ktorom sa práca vypracuje: slovenský
Termín pre odovzdanie práce: 17.04.2025
Dátum zadania diplomovej práce: 31.10.2024



1-2. P. Vokorokos
prof. Ing. Liberios Vokorokos, PhD.
dekan fakulty

Čestné vyhlásenie

Vyhlasujem, že som celú diplomovú prácu vypracoval samostatne s použitím uvedenej odbornej literatúry.

Košice, 15. apríla 2025

.....

vlastnoručný podpis

Pod'akovanie

Ďakujem vedúcemu práce doc. Ing. Stanislavovi Ondášovi, PhD. za jeho odborné vedenie ku danej problematike, usmerňovanie pri písaní diplomovej práce a za poskytnuté materiály ku štúdiu problematiky práce.

Taktiež ďakujem mojim kolegom, ktorí mi pomohli túto prácu otestovať v reálnom čase a dať mi tak rýchlu odozvu.

Obsah

Zoznam obrázkov	12
Zoznam tabuliek	16
Zoznam symbolov a skratiek	17
Úvod	18
1. Logopédia	19
1.1. Definícia logopédie	19
1.2. Delenie logopédie	19
1.2.1. Logopédia pre dospelých	19
1.2.2. Logopédia pre deti	19
1.3. Logopedické poruchy u detí	20
1.3.1. Dyslália	20
1.3.2. Rotacizmus	20
1.3.3. Sigmatizmus	20
1.3.4. Iné druhy dyslálie	20
1.3.5. Ďalšie logopedické poruchy u detí	21
1.4. Logopedické cvičenia pre deti	21
1.4.1. Orofaciálne svalstvo	21
1.4.2. Mimické svalstvo	21
1.4.3. Svaly jazyka a mäkkého podnebia	21
1.4.4. Cvičenie jazyka	21
1.4.5. Budúcnosť logopedických cvičení	22
2. Gamifikácia	23
2.1. Definícia pojmu hra	23
2.2. Definícia pojmu počítačová hra	24
2.2.1. Kategorizácia počítačových hier	24
2.2.2. Ďalšie rozdelenie hier	28
2.3. Definícia gamifikácie	29

2.4.	História gamifikácie.....	29
2.5.	Aktuálne trendy gamifikácie	30
2.6.	Pravidlá gamifikácie.....	31
2.6.1.	Viditeľnosť	31
2.6.2.	Zrozumiteľnosť	32
2.6.3.	Jasný odkaz.....	32
2.6.4.	Atraktivita	32
2.7.	Výhody a nevýhody gamifikácie	32
2.7.1.	Výhody gamifikácie	33
2.7.2.	Nevýhody gamifikácie	33
2.8.	Terapeutické využitie gamifikácie	33
2.8.1.	Gamifikácia pre liečenie duševných porúch.....	34
2.8.2.	Gamifikácia pre liečenie fyzických porúch	35
2.8.3.	Gamifikácia pre rehabilitačné nácviky rúk	35
2.8.4.	Gamifikácia a logopédia	36
2.9.	Budúcnosť gamifikácie	38
3.	Návrh logopedickej platformy LogoHra	39
3.1.	Cieľ práce.....	39
3.2.	Použité technológie.....	39
3.2.1.	HyperText Markup Language	39
3.2.2.	Cascading Style Sheets	40
3.2.3.	Bootstrap.....	40
3.2.4.	JavaScript.....	41
3.2.5.	Phaser 3.....	41
3.2.6.	jQuery a AJAX	42
3.2.7.	Web Speech API	42
3.2.8.	Hypertext Preprocessor	43
3.2.9.	MySQL	43

3.2.10.	XAMPP	44
3.3.	Štruktúra platformy.....	44
3.4.	Návrh a dizajn platformy	46
3.4.1.	Návrh a dizajn navigačnej lišty	46
3.4.2.	Návrh a dizajn úvodnej podstránky.....	47
3.4.3.	Návrh a dizajn administrátorského rozhrania	48
3.4.4.	Návrh a dizajn podstránky správy hier	48
3.4.5.	Návrh a dizajn podstránky správy datasetov	49
3.4.6.	Návrh a dizajn podstránky správy úloh	50
3.4.7.	Návrh a dizajn podstránky správy obrázkov.....	51
3.4.8.	Návrh a dizajn podstránky správy kategórií obrázkov	52
3.4.9.	Návrh a dizajn podstránky zoznamu používateľov.....	53
3.4.10.	Návrh a dizajn podstránky používateľa	54
3.4.11.	Návrh a dizajn podstránky hry	55
3.5.	Vytvorenie databázy platformy.....	56
3.5.1.	Nosné tabuľky	56
3.5.2.	Herné tabuľky	56
3.5.3.	Používateľské tabuľky.....	56
3.6.	Hierarchia súborov	58
3.7.	Implementácia registrácie a prihlasovania	58
3.7.1.	Vytvorenie súboru login.js.....	59
3.7.2.	Vytvorenie formulára pre registráciu.....	59
3.7.3.	Vytvorenie formulára pre prihlásenie	59
3.7.4.	Prepojenie formulárov s backendom	59
3.7.5.	Spracovanie registračných dát	60
3.7.6.	Prihlásenie používateľa	62
3.8.	Implementácia kategórií obrázkov.....	62
3.9.	Implementácia obrázkov	65

3.10.	Implementácia úloh	68
3.11.	Implementácia datasetov.....	70
3.11.1.	Výpis, tvorba a odstránenie datasetov.....	70
3.11.2.	Výpis a odstránenie úloh v datasete	71
3.11.3.	Pridávanie nových úloh do datasetu	72
3.12.	Implementácia hier	73
3.12.1.	Zobrazenie hier.....	73
3.12.2.	Pridávanie nových hier	73
3.12.3.	Úprava a odstránenie hry.....	76
3.12.4.	Priradovanie používateľov	77
3.13.	Implementácia správy používateľov	78
3.14.	Zobrazenie používateľského účtu	79
3.15.	Implementácia úvodnej stránky.....	80
3.16.	Implementácia hry	81
3.16.1.	Predpríprava hodnôt z databázy	81
3.16.2.	Importovanie frameworku Phaser 3 a inicializácia globálnych premien	82
3.16.3.	Funkcia preload().....	82
3.16.4.	Funkcia create().....	83
3.16.5.	Funkcia rollDice().....	85
3.16.6.	Funkcia moveForward()	85
3.16.7.	Funkcia loadImages().....	87
3.16.8.	Funkcia loadRiddle()	88
3.16.9.	Funkcia handleSpeechInput().....	89
3.16.10.	Funkcia saveGameResults().....	90
4.	Finálne úpravy a testovanie	91
4.1.	Vytvorenie hry Classic	91
4.2.	Otestovanie aplikácie	92
	Záver.....	97

Zoznam použitej literatúry	98
Prílohy	101

Zoznam obrázkov

Obr. 1 - Ukážka aplikácie RehaLingo [4]	22
Obr. 2 - Stolná hra Senet	24
Obr. 3 - Ukážka hry Final Fantasy VII Rebirth	25
Obr. 4 - Ukážka hry F1 24	26
Obr. 5 - Ukážka hry Farming Simulator 22	27
Obr. 6 - Ukážka hry Age of Empires II: Definitive Edition	27
Obr. 7 - Dizajn úrovne z hry The Legend of Zelda: The Twilight Princess [11]	28
Obr. 8 - Koncept gamifikácie [23]	30
Obr. 9 - Ukážka hry Kahoot	31
Obr. 10 - Graf využitia gamifikácie pri psychických problémoch [18]	34
Obr. 11 - Reakcia na rôzne pohyby rúk [21]	36
Obr. 12 - Zariadenie Leap Motion [21]	36
Obr. 13 - Ukážka hry Poznávaj aplikácie - Logopédia	37
Obr. 14 - Budúcnosť gamifikácie pomocou VR [22]	38
Obr. 15 - Ukážka štruktúry HTML jazyka	39
Obr. 16 - Ukážka štruktúry CSS jazyka	40
Obr. 17 - Ukážka využitia frameworku Bootstrap v HTML kóde	40
Obr. 18 - Ukážka štruktúry JS jazyka	41
Obr. 19 - Inicializácia Phaser 3 v kóde	41
Obr. 20 - Ukážka využitia knižnice jQuery	42
Obr. 21 - Ukážka volania AJAX pre komunikáciu so serverom	42
Obr. 22 - Princíp fungovania komunikácie klient-server	43
Obr. 23 - Ukážka MySQL príkazov	43
Obr. 24 - Ukážka aplikácie XAMPP	44
Obr. 25 - Štruktúra platformy	45
Obr. 26 - Návrh navigačnej lišty a formulárov na prihlásenie/registráciu	46
Obr. 27 - Dizajn úvodnej podstránky	47
Obr. 28 - Dizajn administrátorského rozhrania	48
Obr. 29 - Dizajn podstránky správy hier	49
Obr. 30 - Dizajn podstránky správy datasetov	50
Obr. 31 - Dizajn podstránky pre správu úloh	51
Obr. 32 - Dizajn podstránky správy obrázkov	52
Obr. 33 - Dizajn podstránky správy kategórie obrázkov	53

Obr. 34 - Dizajn podstránky zoznamu používateľov.....	53
Obr. 35 - Dizajn podstránky používateľa	54
Obr. 36 - Dizajn hry.....	55
Obr. 37 - Vytváranie nového administrátorského kľúča	57
Obr. 38 - Náhľad databázy platformy.....	57
Obr. 39 - Ukážka príkazu CREATE TABLE	58
Obr. 40 - Ukážka vytvorenia vstupného elementu pre meno	59
Obr. 41 - Príklad použitia AJAX.....	60
Obr. 42 - Vytvorenie spojenia s databázou	60
Obr. 43 - SQL príkaz na kontrolu e-mailovej adresy.....	61
Obr. 44 - Neúspešná registrácia s dôvodu existujúceho prihlasovacieho mena	61
Obr. 45 - Overenie kľúča	62
Obr. 46 - Nahratie údajov do tabuľky users	62
Obr. 47 - SQL dopyt na zobrazenie kategórií.....	63
Obr. 48 - Plynulé odstránenie kategórie	63
Obr. 49 - Úprava názvu kategórie	63
Obr. 50 - Zobrazenie novej kategórie po odpovedi na AJAX.....	64
Obr. 51 - Výpis kategórie obrázkov	64
Obr. 52 - Výpis kategórií obrázkov	65
Obr. 53 - Výpis obrázkov	65
Obr. 54 - Vyriešenie problému s načítaním obrázka	66
Obr. 55 - Uloženie nového obrázka.....	66
Obr. 56 - Úprava obrázku	67
Obr. 57 - Odstránenie obrázka	67
Obr. 58 - Ukážka výpisu obrázkov	68
Obr. 59 - Pridanie hádanky do databázy	69
Obr. 60 - Výpis hádanky	69
Obr. 61 - Výpis jednoduchých slov	69
Obr. 62 - SQL dopyt pre vyhľadanie datasetov	70
Obr. 63 - Výpis datasetov	71
Obr. 64 - Ukážka SQL dopytu JOIN SELECT.....	71
Obr. 65 - Obsah datasetu	72
Obr. 66 - Využitie dvojitého SELECT dopytu.....	72
Obr. 67 - Pridanie úlohy do datasetu	72
Obr. 68 - Výpis hry.....	73

Obr. 69 - Editor mapy	74
Obr. 70 - Vytvorená mapa	75
Obr. 71 - Rekurzívna funkcia odstránenia priečinka	76
Obr. 72 - Rekurzívna funkcia kopírovania priečinka	76
Obr. 73 - SELECT dopyty pre aktuálnych/potenciálnych používateľov	77
Obr. 74 - Aktuálni používatelia	77
Obr. 75 - Funkcia vyhľadávania používateľov	78
Obr. 76 - Výpis používateľa	78
Obr. 77 - Zobrazenie používateľa	79
Obr. 78 - Zobrazenie podrobných výsledkov	80
Obr. 79 - SQL dopyt pre výpis špecificky pridelených hier	80
Obr. 80 - Výpis verejných hier	81
Obr. 81 - Načítanie adries pre priečinkov hry	81
Obr. 82 - Importované frameworky Phaser 3	82
Obr. 83 - Ukážka globálnych premien	82
Obr. 84 - Načítanie textúr a audio nahrávok	83
Obr. 85 - Cyklus pre vykreslenie herného poľa	83
Obr. 86 - Vykreslenie hráča	84
Obr. 87 - Ukážka vloženia textu do kontajnera	84
Obr. 88 - Importovanie Web Speech API	85
Obr. 89 - Zobrazenie hodu kockou	85
Obr. 90 - Určenie polohy pohybu	86
Obr. 91 - Vykreslenie obrázkovej úlohy	87
Obr. 92 - Úloha doplnenia slova	88
Obr. 93 - Zobrazenie jednoduchej úlohy	88
Obr. 94 - Zobrazenie a hádanky a pomôcky	88
Obr. 95 - Správna odpoveď	89
Obr. 96 - Nesprávna odpoveď	89
Obr. 97 - Uloženie podrobných výsledkov	90
Obr. 98 - Vygenerovaný obrázok pozadia	91
Obr. 99 - Úvodná stránka	93
Obr. 100 - Vytvorené hry	93
Obr. 101 - Vytvorené datasety	94
Obr. 102 - Pridané obrázky	94
Obr. 103 - Zoznam kategórií	94

Obr. 104 - Vytvorenie používateľa	95
Obr. 105 - Ukážka hry	95
Obr. 106 - Koniec hry.....	95
Obr. 107 - Mobilný dizajn.....	96
Obr. 108 - Mobilný dizajn pre úvodnú podstránku a formuláre	96

Zoznam tabuliek

Tab. 1 - Súbory v administrátorskej časti	58
---	----

Zoznam symbolov a skratiek

AJAX	Asynchronous JavaScript And XML
API	Application Programming Interface
AR	Augmented Reality
CSS	Cascading Style Sheet
DOM	Document Object Model
FPS	First Person Shooter
HTML	HyperText Markup Language
HTTPS	HyperText Transfer Protocol Secure
JPEG	Joint Photographic Experts Group
JS	JavaScript
JSON	JavaScript Object Notation
OS	Operating System
PHP	Hypertext Preprocessor
PNG	Portable Network Graphics
RDBMS	Relational Database Management System
RDP	Remote Desktop Protocol
RPG	Role Playing Game
SQL	Structured Query Language
URL	Uniform Resource Locator
VPN	Virtual Private Network
VR	Virtual Reality
WIN	Windows
WHO	World Health Organization
XML	Extensible Markup Language

Úvod

V súčasnosti existuje mnoho logopedických praktík a cvičení, ktoré majú za úlohu pomôcť dieťaťu s rôznymi logopedickými poruchami. Avšak najnovší trend poukazuje na precvičovanie pomocou rôznych aktivít a hier. Logopedické cvičenia fungujú na princípe návštevy pacienta v logopedickej ambulancii, ak ide o logopedického odborníka alebo v škole, ak sa jedná o školského logopéda. Lekár stanoví diagnózu pacienta a určí logopedické cvičenia, ktoré musí pacient vykonávať, aby bolo možné daný logopedický problém eliminovať. Cvičenia môžu pôsobiť na dieťa stresujúco, pretože prostredie nemusí vyhovovať každému dieťaťu rovnako. Klasické návštevy u logopedického odborníka môžu byť nahradené aplikáciou, ktorú daný odborník môže sledovať a analyzovať výsledky pacienta, ktorý vykonáva danú aktivitu. Odborník následne podľa výsledkov môže stanoviť ďalšie cvičenia alebo navrhnúť ukončenie cvičení, ak sa problém zredukoval alebo eliminoval. Týmto spôsobom môžeme dieťa „donútiť“ alebo motivovať pri vykonávaní logopedických cvičení pomocou rôznych odmien podľa toho, aký výkon podá pri ich vykonávaní.

V súčasnosti sa hry a herné prvky stávajú každodennou súčasťou mnohých ľudí, kde herné prvky neslúžia iba na zábavu, ale aj na edukáciu a vylepšovanie zážitku. Tento termín sa nazýva gamifikácia. Aký je rozdiel medzi gamifikáciou a klasickou hrou? Pod termínom „hra“ rozumieme štandardné typy hier – počítačové hry, stolové hry, športové hry, hry na detskom ihrisku. Hlavným cieľom hier je vytvorenie emócií a hlavne vytvoriť zábavu u daného hráča. Jedným z odvetví, kde je možné uplatniť hry je aj logopédia.

Cieľom práce je priblížiť čitateľovi pojem gamifikácia a následne ukázať vytvorenie nástroja, ktorý môže pomôcť pri riešení logopedických porúch. V prvej kapitole si stručne priblížime pojem logopédia, čo to znamená, aké sú najčastejšie logopedické poruchy u detí a ako je ich možné precvičiť. V druhej kapitole následne nadviažeme na samotný pojem gamifikácia. Rozoberieme čo znamená pojem hra, kedy vznikol a aké sú jeho rozdelenia. Plynule prejdeme na pojem gamifikácia, zodpovieme jej vlastnosti, aké pravidlá by sa mali dodržiavať, výhody a nevýhody, ako je v súčasnosti využívaná a aký potenciálny prínos môže mať v budúcnosti.

V tretej kapitole začneme s tvorbou platformy LogoHra, ktorá umožní vytvorenie hry podľa požiadaviek administrátora. Rozoberieme si aké technológie sme využili pri tvorbe, následne si ukážeme návrhy a dizajny jednotlivých podstránok a na záver samotné implementácie jednotlivých riešení.

V poslednej kapitole si ukážeme vytvorenie demo verzie hry, ktorá bude slúžiť ako ukážka, čo dokáže naša platforma a jej nasadenie do školskej siete, v ktorej budú následne vykonávané testy dizajnu a funkcionality a zhodnotíme výsledky testovania.

1. Logopédia

Aby sme pochopili významu gamifikácie v oblasti logopédie, musíme si najprv vysvetliť logopedickú oblasť, pre ktorú chceme vytvoriť aplikáciu.

1.1. Definícia logopédie

Logopedické problémy vznikajú pri výslovnosti či komunikácií, no u niektorých detí ale aj dospelých ľudí sa môžu prejaviť rôzne ťažkosti, či už v dôsledku vrodených predispozícií alebo vonkajších faktorov. Logopédia je disciplína, ktorá pomáha odstraňovať alebo zmierňovať problémy výslovnosti pomocou špeciálnych cvičení a terapií. Odborník v logopedickej disciplíne sa nazýva logopéd, ktorý diagnostikuje poruchy reči a jazyka a navrhuje individuálny terapeutický plán pre každého pacienta. Diagnostika prebieha formou sedenia, ktoré môže trvať v rozsahu 20 až 90 minút v závislosti od závažnosti problému. Počas diagnostiky logopéd analyzuje rôzne aspekty reči a snaží sa vytvoriť uvoľnené prostredie, aby výsledky neboli skreslené stresom. Potenciálne môže byť diagnostika rozdelená do niekoľkých sedení, aby sa presnejšie diagnostikoval problém [1][2].

1.2. Delenie logopédie

Logopédia sa delí na dve základné kategórie podľa veku človeka. Patrí tu logopédia pre dospelých a pre deti.

1.2.1. Logopédia pre dospelých

Logopédia pre dospelých má za úlohu pomôcť s poruchami reči pre ľudí starších ako 18 rokov. Poruchy u dospelých osôb vznikajú ako sekundárny efekt, kde najčastejšími príčinami sú poškodenia mozgových buniek vnútorným krvácaním, mozgovým nádormi alebo vonkajšími vplyvmi ako sú rôzne vážne dopravné nehody alebo ťažké úrazy spôsobené pádom na oblasť hlavy. Odborníci v oblasti logopédie pre dospelých sa nazývajú klinickí logopédi [1][2].

1.2.2. Logopédia pre deti

V oblasti detskej logopédie poznáme mnoho rôznych porúch reči a jazyka. Poruchy môžu byť zapríčinené mnohými faktormi. Spektrum porúch vieme rozdeliť na vrodené, psychické a environmentálne. Vrodené poruchy súvisia so zle vyvinutými rečovými orgánmi, kde môžu byť poruchy anomálne a po istom čase zaniknúť, ale ak nie sú na ústupe, je nutné vyhľadať klinického logopéda. Pri psychických problémoch môže nastať porucha výslovnosti pri traume, kde si dieťa vybuduje istý psychický blok a pri určitom slove alebo komunikácii si to spájať s danou nepríjemnou udalosťou. Posledným prípadom sú environmentálne poruchy, kde dieťa získava zlú výslovnosť z prostredia, v ktorom žije. Pri nebiologických poruchách môžu rodičia dieťaťa navštíviť školského logopéda. Pri detských poruchách je najdôležitejšie logopedický problém začať riešiť čo najskôr [1].

1.3. Logopedické poruchy u detí

Existuje mnoho logopedických porúch, ktoré môžu postihovať ľudí rôznych vekových kategórií. Najčastejšou logopedickou poruchou u detí je dyslália [1].

1.3.1. Dyslália

Dyslália sa vyskytuje prevažne vo vekovom rozmedzí do 7. roku života dieťaťa. Charakteristickou črtou dyslálie je zlá výslovnosť jednej alebo viacerých hlások v rodnom jazyku. Deti s dysláliou môžu hlásky nesprávne artikulovať alebo ich zamieňať za iné, čo vedie k nejasnej či nezrozumiteľnej reči. Dyslália má vývinový charakter, pretože deti často napodobňujú nesprávne vyslovené slová, najmä tie, ktoré počujú od rodičov. Problém sa dá postupne odstrániť, najmä v školskom veku alebo návštevou školského logopéda, ak problém s výslovnosťou pretrváva dlhšie. Existuje niekoľko typov dyslálie, ktoré vieme zdefinovať nesprávnou výslovnosťou konkrétnych písmen [1].

1.3.2. Rotacizmus

Rotacizmus je druh dyslálie, ktorého výskyt je najčastejší. Príznak, ktorým sa rotacizmus prejavuje je nesprávne vyslovovanie hlások „R/Ř“ v slovenskom a českom jazyku. Rotacizmus je taktiež delený podľa toho, ako sa hláska nesprávne vyslovuje. Zamieňaním hlásky „R“ za inú hlásku, napríklad hlásku „L“ označujeme ako pararotacizmus. Vynechávanie hlásky „R“ sa označuje ako mogirotacizmus. Najznámejším rotacizmom je ráčkovanie alebo odborne nazývaný velárny rotacizmus [1].

1.3.3. Sigmatizmus

Sigmatizmus ako ďalší druh dyslálie sa vyznačuje nesprávnym vyslovovaním sibilanov (hlások z kategórie sykaviek), kde patria hlásky „S/Š“, „C/Č“, „Z/Ž“, „DZ/DŽ“. Podobne ako pri rotacizme, aj pri sigmatizme poznáme niekoľko delení. Pod pojmom predný sigmatizmus poznáme zámenu hlások „S“ a „Z“ za hlásky mäkkej varianty „Š“ a „Ž“. Ďalšie varianty sigmatizmu sú parasigmatizmus, prejavujúci sa zamieňaním sykaviek, zadný sigmatizmus, ktorý podobne ako predný sigmatizmus spôsobuje zámenu hlások „S“ a „Z“, ale za hlásky „Š“, „Č“, „Ň“, „Ť“. Ako najznámejší zástupca sigmatizmu je považovaný interdentalný sigmatizmus, ktorý sa laicky nazýva ako šušľavosť [1].

1.3.4. Iné druhy dyslálie

Medzi menej časté druhy dyslálie patrí kappacizmus, ktorý sa prejavuje nesprávnou výslovnosťou hlásky „K“, a lambdacizmus, spôsobujúci chybné vyslovovanie hlásky „L“. Ďalšími typmi sú gammacizmus, ktorý ovplyvňuje hlásku „G“ a chitizmus, pri ktorom dochádza k nesprávnej artikulácii hlásky „CH“. Okrem nich rozlišujeme aj tetacizmus a detacizmus, ktoré vedú k nesprávnej

výslovnosti hlások „T“ a „D“, ako aj jotacizmus, prejavujúci sa problémami pri artikulácii hlásky „J“.

[1].

1.3.5. Ďalšie logopedické poruchy u detí

Okrem dyslálie existujú aj ďalšie logopedické poruchy u detí, ako napríklad dysfázia, ktorá ovplyvňuje vývin reči, dyslexia, spôsobujúca ťažkosti s čítaním a písaním, koktanie, ktoré narúša plynulosť reči, mutizmus, pri ktorom dieťa odmieta hovoriť v určitých situáciách, dysfónia, vedúca k poruchám hlasu, a afónia, ktorá sa prejavuje úplnou stratou hlasu. Poruchy môžu mať rôzne príčiny a vyžadujú špecifické logopedické prístupy [1][2].

1.4. Logopedické cvičenia pre deti

Logopedické cvičenia pomáhajú deťom zdokonaľovať výslovnosť hlások a slov. Ich cieľom je aktivácia orofaciálneho, mimického svalstva a svalov jazyka, čím sa zlepšuje artikulácia. Precvičovaním týchto svalstiev si dieťa zdokonaľuje jemnú motoriku svalov, a tým potenciálne dokáže pomaly eliminovať logopedický problém [1][2].

1.4.1. Orofaciálne svalstvo

Orofaciálne svalstvo zahŕňa svaly úst, tváre a plní funkcie: mimickú (výraz tváre), mastifikačnú (žuvanie), fonačnú (tvorba hlasu) a stabilizačnú (hlava a krčná chrbtica). Cvičenia sa vykonávajú minimálne päťkrát denne, symetricky a bez nadmerného napätia. Orofaciálne cvičenia zahŕňajú nácviky artikulačného výdychu a nácvik čelústneho uhlu [1].

1.4.2. Mimické svalstvo

Mimické svaly sa delia na čelnú (výraz tváre), očnú (pohyb viečok) a ústnu skupinu (artikulácia a žuvanie). Cvičenia môžu zahŕňať uvoľňovanie (relaxácia svalstva) alebo posilňovanie (udržiavanie pozícií 3 až 5 sekúnd). Medzi cvičenia môžeme zaradiť zdvíhanie obočia, špúlenie pier, otváranie úst či pohyb kútikov úst [1].

1.4.3. Svaly jazyka a mäkkého podnebia

Jazyk je najpohyblivejší sval tela a jeho pohyb ovplyvňujú vlastné a vonkajšie svaly. Jazykom je možné vykonávať rôzne ohybné a motorické pohyby. Mäkké podnebie zohráva úlohu pri prehltaní a tvorbe zvukov. Precvičovanie mäkkého podnebia môže byť vykonávané aj pomocou jazyka [1].

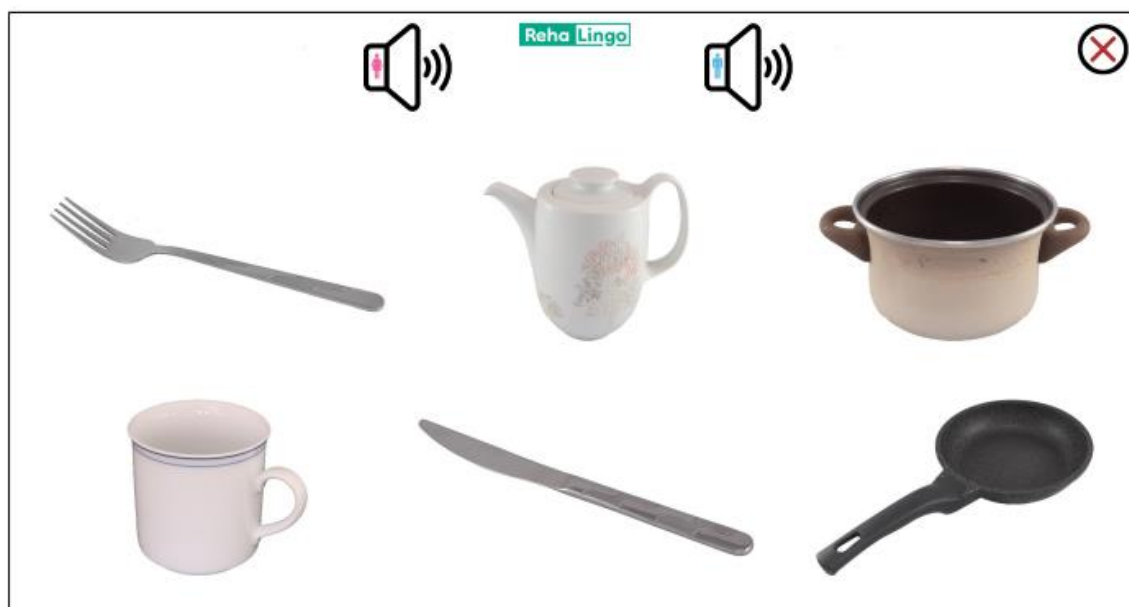
1.4.4. Cvičenie jazyka

Cvičenia prebiehajú v troch fázach: relaxácia (správna poloha jazyka), uvoľňovanie (vyplazovanie a pohyb jazyka) a posilňovanie (tlak jazyka proti objektom). Posilňovanie sa vykonáva s prstom alebo iným predmetom, pričom sa precvičuje oslabená časť jazyka. Preťahovanie jazyka sa môže vykonávať aj s chladenými predmetmi, napr. zmrzlinou [1].

1.4.5. Budúcnosť logopedických cvičení

Súčasným trendom pri riešení problémov aj v oblasti logopédie je pojem nazývaný gamifikácia. V budúcnosti by mohla gamifikácia priniesť personalizované terapeutické programy, ktoré by sa prispôbovali individuálnym potrebám každého pacienta, čím by sa zvýšila efektivita terapie. Výskum vo vzdelávacej oblasti naznačuje, že integrácia herných prvkov do vzdelávania a terapií môže viesť k lepším výsledkom a vyššej spokojnosti účastníkov.

V článku "RehaLingo – Towards a Speech Training System for Aphasia" autori predstavujú koncept počítačového systému na podporu rečovej terapie pre pacientov s afáziou. Súčasťou systému je aj gamifikačný prístup, ktorý má za cieľ zvýšiť motiváciu a zapojenie pacientov do terapeutického procesu. V rámci gamifikácie sa plánuje implementácia rôznych herných prvkov, ako sú bodové hodnotenia, úrovne obťažnosti a vizuálne odmeny za správne odpovede. Herné prvky majú podporiť pacientov v pravidelnom cvičení a zlepšiť efektivitu terapie. Autori tiež navrhujú použitie videoklipov terapeuta alebo avatara, ktorý by poskytoval spätnú väzbu a rady pri nesprávnych odpovediach, čím by sa simulovala interakcia s reálnym terapeutom. Cieľom je vytvoriť užívateľsky prívetivé prostredie, ktoré kombinuje terapeutické ciele s prvkami hry na zlepšenie jazykových schopností pacientov s afáziou [4].



Obr. 1 - Ukážka aplikácie RehaLingo [4]

Gamifikácia má celkový potenciál stať sa významným nástrojom v logopedickej terapii, ktorý by mohol zlepšiť výsledky liečby a spraviť proces terapie príjemnejším pre deti. V nasledujúcej kapitole bližšie špecifikujeme čo je to gamifikácia, ako súvisí s pojmom hra a taktiež jej uplatnenie v súčasných aplikáciách a terapiách. Na záver kapitoly povieme niečo o jej budúcnosti.

2. Gamifikácia

Na začiatok musíme zadať čo gamifikácia znamená a aký je rozdiel medzi hrou a gamifikáciou z pohľadu používateľa.

2.1. Definícia pojmu hra

Definície hry sú väčšinou založené na vymenovaní jej vlastností. Zvyčajne nie je charakterizovaná jediným znakom, ale kombináciou viacerých rozličných črt. Odborníci, ktorí sa zaoberali hrami a pokúšali sa pojem hry presne vymedziť, sa sústreďovali najmä na opis vlastností a charakteristík hier a hrania. Rôzne atribúty hry, ktoré spomínali autori, možno zhrnúť do nasledujúcich piatich bodov [5]:

- **Hranie hry je dobrovoľné** – Hru si sami vyberáme a sami ju aj riadime.
- **Hra je činnosť** – Samotné hranie považujeme za hodnotnejšie než jej výsledok.
- **Hra má svoje pravidlá** – Pravidlá sú vopred stanovené a nemenia sa počas hry. V prípade neformálnych hier môžu hráči pravidlá upravovať alebo prispôbovať na základe vzájomnej dohody.
- **Hra predstavuje umelo vytvorenú situáciu** – Je do istej miery oddelená od reálneho života a často si vyžaduje predstavivosť a predvídavosť hráčov.
- **Hra je interaktívna** – Vyžaduje aktivitu a pozornosť hráča, ale pritom nesmie vytvárať stresové situácie pre hráča [5].

Hra môže zahŕňať aktivity jednotlivca, dvojice, malej alebo veľkej skupiny poprípade celej komunity, ak sa jedná o veľkú udalosť. Niektoré hry vyžadujú na hranie hier špeciálne pomôcky. Väčšina hier je založená na sociálnej interakcii s jasne definovanými pravidlami. Počas hry sa kladie veľký dôraz na jej riadenie a priebeh [5].

Východisková situácia, priebeh a výsledky niektorých hier sa dajú analyzovať a presne skúmať. Pravidlá hry a rozhodovanie jej účastníkov možno formalizovať, riadiť a optimalizovať. Taktiež je možné vytvárať algoritmy pre výherné stratégie. To pri niektorých hrách umožňuje kreatívne myslenie, kde správna taktika alebo stratégia umožní víťazstvo nad oponentom, či už virtuálnym v podobne hernej postavy (tzv. NPC – Non Playable Character) alebo reálnym oponentom, ktorý môže byť s hráčom v danej miestnosti alebo lokalizovaný na inom mieste. Medzi najstaršie hry na svete patrí šach, ktorý sa datuje do doby Starovekej Indie a taktiež hry Senet zo Starovekého Egypta alebo Go zo Starej Číny. Hry sú postavené na hre dvoch hráčov, ktorí súperia proti sebe a hracia plocha pozostávala z dosky, na ktorej sú umiestnené rôzne druhy figúrok [5].



Obr. 2 - Stolná hra Senet

2.2. Definícia pojmu počítačová hra

Počítačová hra je typ hry, ktorá vyžaduje na hranie výpočtové zariadenie (počítač, mobil, herná konzola, handheld). Patrí medzi najrozšírenejšie hry súčasnosti, kde začiatky sa datujú do roku 1962, kedy bola vytvorená prvá „počítačová hra“ Spacewar!. V 70. rokoch 20. storočia nastal obrovský vývoj nových počítačových hier, kde trend pokračoval až do súčasnosti. Počítačové hry sa líšia od klasických stolových alebo iných hier tým, že dokážu ponúknuť zážitok, ktorý sa podobá sledovaniu filmu, ale s tým rozdielom, že hráč je centrom diania hry a rozhoduje ako bude postupovať v jeho príbehu. Taktiež ponúkajú prakticky neobmedzenú kreativitu a rôzne štýly hrania, kde pri stolových hrách z minulosti mali často minimálny až žiadny príbeh a fungovali na princípe ťahovej stratégie (1 ťah = 1 pohyb hráča).

V oblasti počítačových hier ich rozdeľujeme do rôznych typov kategórií, podľa ktorých by sme ich vedeli rozdeliť. V súčasnosti počítačové hry sa vyvinuli do štádia, kedy vznikol tzv. e-šport, kde hráči súťažajú v turnajoch o rôzne ceny a tituly majstrov sveta v danej hre. V roku 2025 sa dokonca začnú prvé olympijské hry v e-športe v Saudskej Arábii pod názvom Olympic Esports Games [7].

2.2.1. Kategorizácia počítačových hier

Prvou kategóriou je rozdelenie podľa žánru. Kategorizácia podľa žánru je najrozšírenejšie delenie, kde sa určuje tematika danej hry, pre koho je určená a aké typické prvky daná hra obsahuje. Toto rozdelenie je najkomplexnejšie pretože existuje veľký počet žánrov, kde mnohé súčasné hry prenikajú do niekoľkých žánrov. V ďalšej časti popíšeme niektoré známe herné žánre.

2.2.1.1. Akčné hry

Akčné hry sú žáner, ktorých hlavnou úlohou je vyvolať adrenalín alebo vzbudiť u hráča rýchle reakcie na rýchlo meniaci sa dej hry. Pri akčných hrách nie je primárnym cieľom príbeh hry, ale skôr reakcia hráča. Do kategórie akčných hier môžeme zaradiť rôzne typy hier napr. strelačky označované ako FPS (First Person Shooter), ale taktiež rôzne bojové hry. Taktiež v minulosti ku akčným hrám patrili plošinové hry ako napríklad Pac-Man, Super Mario, Space Invaders, a ďalšie [5][6].

2.2.1.2. RPG Hry

RPG (Role Play Games) hry sa sústreďujú primárne na celkový zážitok z hry. Primárny cieľ RPG žánra je ovládať postavu v záujme hráča. V rámci kontroly postavy je možné vylepšovať postavu, používať rôzny bojový štýl, rozhodovať o deji príbehu (ak túto možnosť umožňuje hra) podľa záľuby hráča. V týchto hrách je dej a príbeh veľmi dôležitou časťou. Prevláda v nich otvorený svet (open world), kde hra dáva voľnosť hráčovi čo konkrétne robiť alebo preskúmať. Vo väčšine prípadov sa RPG hry hrajú z pohľadu tretej osoby (TPW – Third Person View). RPG hry majú taktiež mnoho vedľajších úloh, ktoré majú zabezpečiť predĺženie herného zážitku alebo rozšírenie príbehu (lore) hry. Celkovo, čo sa týka hrateľnosti patrí RPG žáner medzi najdlhšie trvajúce herné zážitky (kvalitná RPG hra by mala mať vyše 40 hodín na dokončenie príbehu a vyše 100 hodín z hľadiska dokončenia celkovej hry na 100%). RPG žáner patrí medzi najrozsiahlejší herný žáner zo všetkých [5][8].



Obr. 3 - Ukážka hry Final Fantasy VII Rebirth

2.2.1.3. Pretekárske hry

Žáner pretekárskych hier je definovaný motoristickými športami, kde sa snaží napodobniť reálne pretekárske závody ako v skutočných pretekoch. Poznáme dva základné typy pretekárskych hier: simulátory a arkády. Simulátory sú navrhnuté tak, aby hra a herné podmienky pôsobili realisticky a

autenticky, ako sa vozidlo správa, na akom teréne sa nachádza, aká fyzika vplýva na prostredie a pod. Opakom simulátora je arkáda, ktorá je určená pre tých, čo si chcú užiť závod bez veľmi realistického prostredia pre číru zábavu. Ako vyplýva zo žánru príbeh prakticky neexistuje alebo je veľmi minimálny.

Dôležitou úlohou v žánri je súboj hráča buď už proti virtuálnemu protihráčovi (počítaču) alebo proti reálnemu hráčovi poprípade hráčom. Niektoré novšie hry už ponúkajú aj otvorený svet, kde si hráč môže zajazdiť po krajine bez limitácie prostredím alebo pretekárskou dráhou [5][9].



Obr. 4 - Ukážka hry F1 24

2.2.1.4. Simulátory

Simulátory patria ku žánru, ktorý sa snaží priniesť reálne životné situácie, ktoré bežne hráč nemôže reálne zažiť alebo chce vyskúšať aké by to bolo. Niektoré simulátory sa snažia dopodrobna priniesť realistický zážitok s rôznymi možnosťami, mapami alebo nastaveniami. Niektoré len simulujú približne ako funguje daná situácia. Väčšinou preto žáner nemá žiadny dopredu naprogramovaný príbeh, ale necháva na hráčovi ako bude postupovať (vytvára tzv. sandbox - pieskovisko). Simulátory patria medzi viac relaxačný typ hier z dôvodu, že nikde človeka netlačia a taktiež je mu dopriata voľnosť.

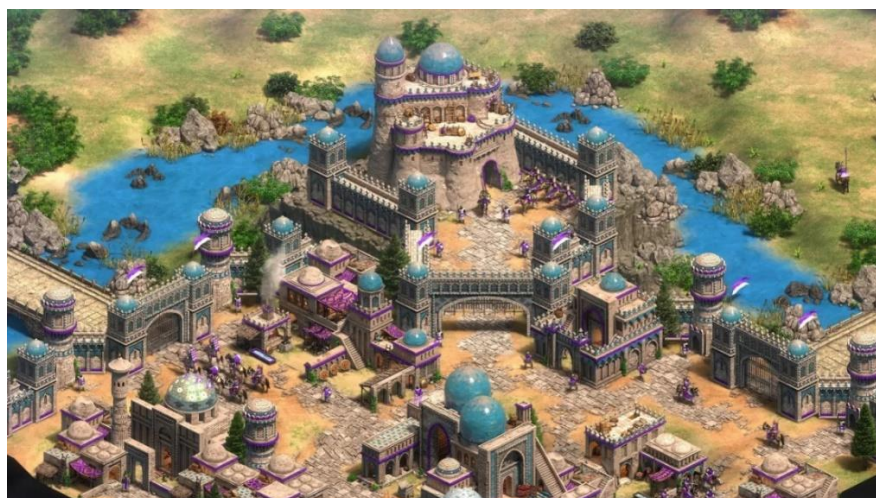
Moderné simulátory ponúkajú veľmi realistickú grafiku vďaka vývoju v oblasti grafických kariet. Pomocou realistickej grafiky dokážu navodiť čiže lepšie zážitok. Niektoré simulátory podporujú funkciu VR, ktorá približuje hráča na tenkú hranicu medzi realitou a hrou. Medzi najväčšie hry žánru patria série Farming Simulator, Euro Track Simulator, Flight Simulator a Train Simulator [5].



Obr. 5 - Ukážka hry Farming Simulator 22

2.2.1.5. Strategické hry

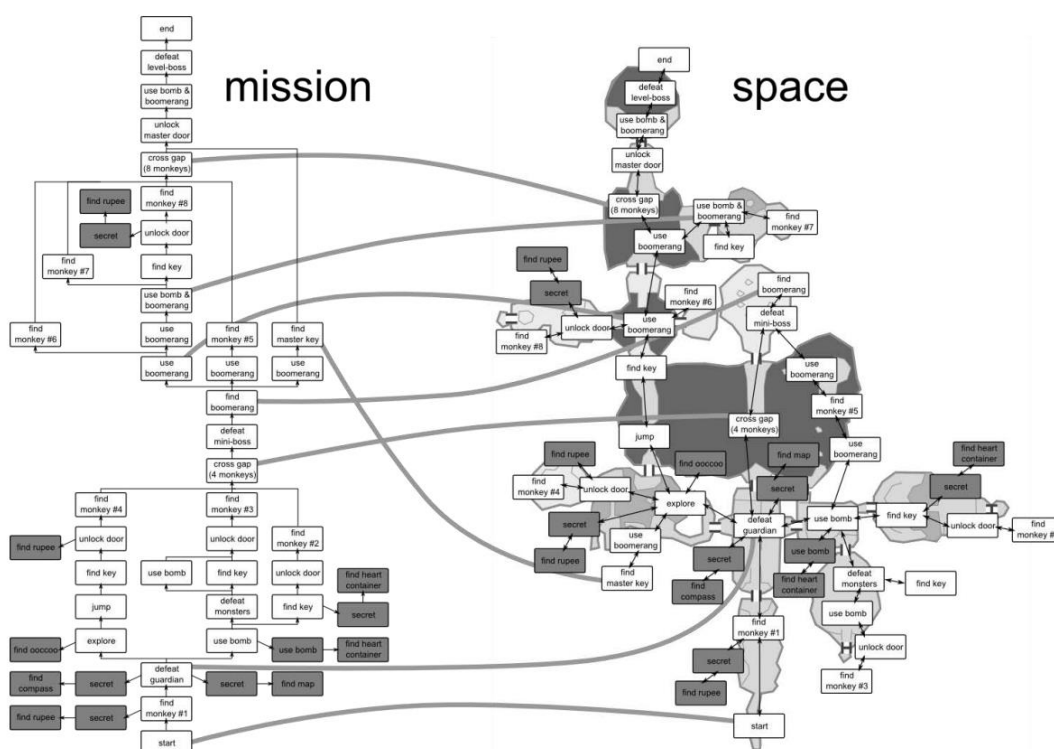
Strategické hry vyžadujú najväčšie množstvo premýšľania. Ako sám názov hovorí, v týchto hrách je najdôležitejšia správne zvolená stratégia. Stratégie sú prevažne ovládané pohľadom zhora na mapu alebo herné pole a hráčovou úlohou je rôznymi príkazmi docieľiť dominanciu na hernom poli. Nie každá strategická hra musí byť bojová. Existujú rôzne strategické hry z pohľadu stavania miest, kde hráč nebojuje proti žiadnemu oponentovi, ale stavia mesto, kde musí brať ohľad na rôzne podmienky. V strategických hrách býva umelá inteligencia na vysokej úrovni a tak pri zvolení vysokej obťažnosti môže pre hráča predstavovať skutočnú výzvu, akoby hral proti rovnako skúsenému hráčovi [5][10].



Obr. 6 - Ukážka hry Age of Empires II: Definitive Edition

2.2.1.6. Adventúrne hry

Adventúry sú žánrer hier, ktoré majú za úlohu prilákať hráča na rôzne dobrodružné elementy hry a majú často v sebe prvky tajomna. Po celej hre môžu byť poskrývané rôzne tajomstvá, ktoré hráč nemusí získať, ale slúžia ako odmena, ak hráč vyrieši rôzne druhy hádaniek, ktoré predchádzajú tomuto tajomstvu. Môžu obsahovať akčné elementy, ale nie je to podmienka. Môžu obsahovať kvalitný príbeh, ale pri adventúrach nie je kladený dôraz na dej alebo príbeh ako napríklad pri RPG hrách. Taktiež nie je zvykom umožňovať rôzne úpravy postavy a často neumožňujú hrať s viacerými hráčmi. Medzi najznámejšie adventúry patrí herná séria Tomb Raider. Na Obr. 7 môžeme vidieť zrealizovaný dizajn úrovne v dvoch rovinách – rovina úloh a ako nasledujú a rovina priestoru, kde sú na mape rozložené úlohy a rôzne tajomstvá [5][11].



Obr. 7 - Dizajn úrovne z hry The Legend of Zelda: The Twilight Princess [11]

2.2.2. Ďalšie rozdelenie hier

Ďalšou kategóriou môže byť rozdelenie podľa interakcie s inými hráčmi (hra pre jedného hráča, kooperačná hra, hra s viacerými hráčmi). V minulosti sa často hry rozdeľovali aj podľa platformy, na ktorú boli vydávané (Playstation-only, Xbox-only, PC-only). V dnešnej dobe tento trend pomaly opadá a tvorcovia hier sa snažia vydávať ich hry na čo najviac platforiem, aby uspokojili čo najviac hráčov na rôznych zariadeniach. Novodobým trendom je prerábanie starých hier do novej grafiky a upravenej hernej logiky. Poznáme tzv. remaster a remake hry, kde remaster najčastejšie znamená iba vylepšenie grafických textúr a oprav chýb a remake úplné prerobenie celej štruktúry hry.

2.3. Definícia gamifikácie

Význam gamifikácie spočíva vo využití herného dizajnu alebo herných elementov pri riešení problematiky v reálnom živote. Herné elementy môžu byť napríklad získavanie hodnotenia buď vo forme počtu bodov (skóre) alebo počtu hviezdíček. Gamifikácia je skvelý spôsob, ako sa naučiť niečo nové, najmä preto, že ponúka oveľa zábavnejší zážitok ako tradičné učenie. Učenie zábavnou formou pôsobí na ľudský mozog oveľa otvorenejšie a ochotnejšie prijímať nové informácie.

2.4. História gamifikácie

Za začiatky gamifikácie môžeme považovať 80. roky minulého storočia, kedy sa objavili dnes už kultové videohry ako Pac-Man alebo Super Mario. Vedci skoro zistili, že hry ovplyvňujú správanie ľudí. Za praotca gamifikácie považujeme Charlesa Coonradta, ktorý bol zakladateľom poradenskej spoločnosti Game of Work a v roku 1984 vydal knihu s rovnakým názvom, v ktorej hľadal odpovede na otázky [12]:

- Aký je rozdiel medzi hobby a zamestnaním?
- Prečo hobby ľudí viac motivuje?

Nechápal, prečo ľudia nezostanú po skončení pracovnej doby na pracovisku, keď dostanú za to plat, ale sa ponáhľajú domov, aby sa mohli venovať svojej záľube a koníčkovi. Zistil, že za túto činnosť, ktorá je zdrojom „flow“ vedomia, sú ľudia ochotní aj platiť. Hobby opísal ako činnosť, ktorá [12]:

- má jasné, dobre ohraničené ciele,
- má dobre merateľný výkon,
- vždy má spätnú väzbu,
- je dobrovoľná.

Gamifikácia ako pojem sa prvý raz použil v prednáške Art of Game Jessie Schella, ktorý načrtol budúcnosť, v ktorej bude každá činnosť gamifikovaná. Zastával názor, že hra nie je len bezcieľna zábava, ale je aj aktívnym nástrojom tvorby hodnôt. Hra je už dávno súčasťou pedagogickej činnosti. Pri gamifikácii na rozdiel od hry, kde je hlavným cieľom zábava, je hra len nástrojom na dosiahnutie cieľov, ktorými sú v prvom rade zvýšenie pozornosti a motivácia [12].

Podľa Kevina Werbracha a Dana Huntera je pri zavádzaní gamifikácie kľúčové najskôr určiť hlavný cieľ. Môže ísť napríklad o zvýšenie pozornosti alebo aktivity. Odporúča sa sústrediť len na jeden cieľ, aby bolo možné lepšie vyhodnotiť jeho dosiahnutie. Ďalším krokom je identifikácia cieľovej skupiny, kde až potom je možné vybrať vhodné nástroje na zlepšenie zážitku z učenia [12].

Zavádzanie gamifikácie prebieha v troch fázach, ktoré si môžeme ilustrovať na vzťahu učiteľa a študenta. Prvá fáza je najnáročnejšia, pretože študenti musia vystúpiť zo svojej komfortnej zóny a prijať nový spôsob učenia. V druhej fáze sa s novou metódou bližšie oboznamujú, postupne si ju osvojujú a začínajú ju pravidelne využívať. Tretia fáza predstavuje štádium, v ktorom už študenti metódu efektívne uplatňujú. Každá z úrovní vyžaduje zvládnutie náročnejších úloh, akými sú napríklad testy. Keďže študenti napredujú rôznym tempom podľa svojich schopností a vedomostí, úlohou učiteľa je poskytovať im podporu a usmerňovať ich smerom k dosiahnutiu stanoveného cieľa [12].



Obr. 8 - Koncept gamifikácie [23]

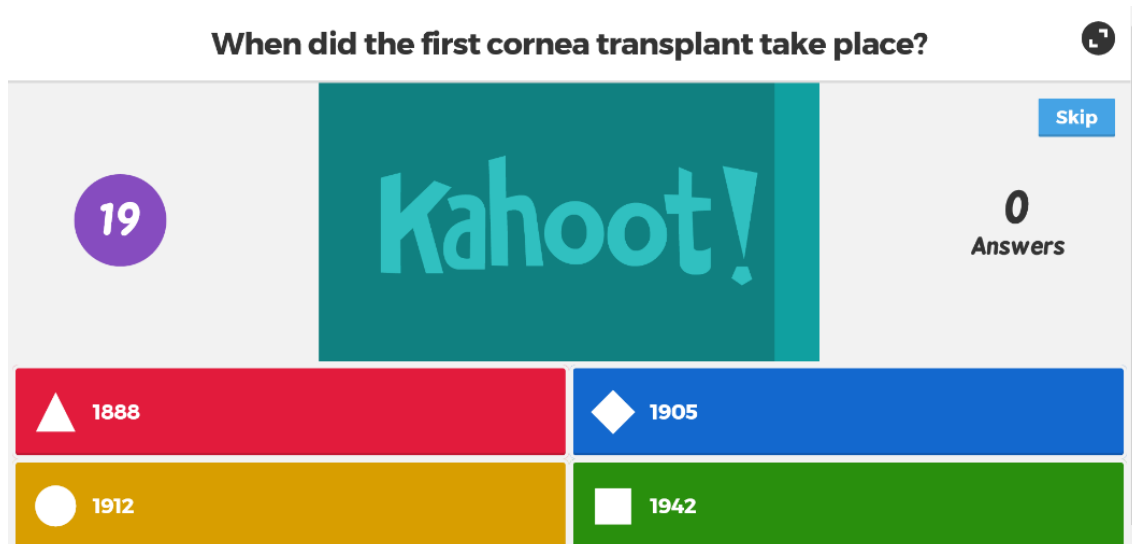
2.5. Aktuálne trendy gamifikácie

Gamifikácia sa dnes stala prirodzenou súčasťou vzdelávacieho procesu v rôznych inštitúciách. Hoci ju možno využiť vo všetkých formách vzdelávania, najčastejšie sa spája s online výučbou. Herné prvky nachádzajú uplatnenie aj vo firemnom vzdelávaní, kde pomáhajú zvyšovať záujem, motiváciu, podporujú testovanie či poskytujú asistenciu. Kľúčové je však ich správne a efektívne začlenenie. V oblasti vzdelávania zároveň vzniká množstvo aplikácií a online platforiem, ktoré sú postavené práve na princípoch gamifikácie [13]:

- **Duolingo** – Aplikácia na učenie a rozširovanie slovnej zásoby rôznych jazykov.
- **Kahoot** – Aplikácia pre vytváranie rôznych vedomostných kvízov.
- **Khan academy** – Aplikácia pre vzdelávanie rôznych vedných odborov.
- **Codecademy go** – Aplikácia pre výučbu programovania a softvérového vývoja.

- **Actionbound** – Aplikácia na podporu tímovej spolupráce pri preberaní či overovaní vedomostí.
- **Minecraft: Education Edition** – Špeciálna verzia známej hry Minecraft, ktorá je prispôbená na vzdelávacie účely a podporuje rozvoj logického myslenia, tímovej spolupráce a programovania.
- **Classcraft** – Platforma, ktorá transformuje tradičné vyučovanie na RPG hru, kde študenti plnia úlohy, získavajú skúsenosti a vylepšujú svoje postavy [13][14].

Nepriamym nástrojom gamifikácie môžeme považovať aj klasické počítačové hry, ktoré sú veľmi dobrým nástrojom na vylepšenie anglického jazyka. Takmer všetky hry sú dnes v anglickom jazyku a to umožňuje zlepšenie slovnej zásoby a potenciálne zlepšenie výslovnosti niektorých slov vďaka tzv. cutscénom. Taktiež niektoré hry obsahujú rôzne hádanky, ktoré môžu vplývať na správny vývoj logického premýšľania, ktoré môže byť použité pri riešení reálnych problémov.



Obr. 9 - Ukážka hry Kahoot

2.6. Pravidlá gamifikácie

Pri gamifikácii musia byť dodržané určité pravidla, aby sa zachovala podstata, ktorú chceme docieľiť.

2.6.1. Viditeľnosť

Prvým pravidlom je viditeľnosť. Viditeľnosť znamená, že je potrebné docieľiť, aby užívateľ presne vedel aká je jeho úloha, aké sú pravidlá a taktiež na akú tému sa cieľi. Ak sa jedná o kreslené pozadie alebo kreslenú tematiku, je najlepšie, aby farby pozadia boli tmavšie alebo mali nižší jas. Okraje objektov alebo postáv je najlepšie ohraničiť čiernou farbou, aby vznikol kontrast medzi každým jedným objektom. Následne farebná výplň by mala obsahovať aspoň jednu výraznú farbu na upútanie užívateľa, avšak nie je vhodné prekombinovať mnoho farieb [13].

2.6.2. Zrozumiteľnosť

Druhým pravidlom je zrozumiteľnosť. Pravidlo hovorí, že užívateľ musí správne pochopiť o akú úlohu sa jedná. Je potrebné mu poskytnúť dostatočný priestor a čas, pri ktorom mu aplikácia povie ako má postupovať pri riešení úlohy. Samozrejme pri niektorých úlohách je možné zadať časový limit, ktorý umožní používateľovi prečítať alebo pochopiť úlohu presne a jasne. Ďalšou vlastnosťou je jednoduchosť, kde nie je aplikácia prekombinovaná a nevytvára zbytočný chaos pri používaní. Nie je preto vhodné používať veľa rýchlych animácií alebo niekoľko animácií naraz, ktoré by narušili sústredenie používateľa [13].

2.6.3. Jasný odkaz

Tretím pravidlom je jasný odkaz. Je dôležité, aby užívateľovi bola sprostredkovaná hlavná myšlienka, téma alebo problematika, ktorú si dokáže zapamätať a použiť v reálnej situácii. Je dôležité, aby boli text a obrázky v čitateľnej forme a nebol použitý bizarný typ písma, pretože v tom by mohol používateľ stratiť hlavnú myšlienku, ktorú si má zapamätať [13].

2.6.4. Atraktivita

Štvrtým pravidlom je atraktivita a príťažlivosť. Na zvýšenie pozornosti a ľahšieho zapamätania posolstva alebo myšlienky je vhodné vytvoriť aplikáciu tak, aby čo najviac upútala pozornosť. Ak sa vytvára aplikácia pre malé deti, je odporúčané používať kreslené obrázky a rôzne postavičky typu pekných priateľských zvierat. Taktiež by odmena pri detských úlohách mala byť viac obrázková ako číselná (napríklad zbieranie hviezdčiek namiesto štandardného počtu bodov/skóre). Pre dospelých sa snaží klásť dôraz skôr na problematiku, na ktorú sa vytvára aplikácia. Taktiež je dôležité, aby používateľské rozhranie bolo čo najjednoduchšie, pretože ľudia stratia záujem, ak je aplikácia veľmi komplikovaná a voľba niektorých možností môže byť zložitá alebo je potrebné zadať veľa krokov [13].

2.7. Výhody a nevýhody gamifikácie

Gamifikácia je silným nástrojom vzdelávania, najmä v hybridných vzdelávacích prostrediach, kde spája tradičné a digitálne metódy učenia. Jej základ spočíva v aktívnom zapojení detí, študentov aj dospelých účastníkov do vzdelávania. Použitie herných prvkov môže zvýšiť motiváciu a podporiť efektívnejšie učenie. V nasledujúcich dvoch bodoch priblížime výhody a nevýhody gamifikácie, ktoré je potrebné brať do úvahy.

2.7.1. Výhody gamifikácie

Gamifikované učenie zvyšuje zapojenie študentov rôznymi spôsobmi [15]:

- **Zvyšuje motiváciu a angažovanosť** – Herné prvky, ako sú odmeny, úrovne a súťaže, môžu podporiť vnútornú aj vonkajšiu motiváciu študentov. Pomáha študentom lepšie porozumieť zložitému materiálu a zachovať si tieto vedomosti.
- **Zvyšuje záujem o učenie vďaka zábavnému a interaktívnemu učeniu** – Kombinácia hier a vzdelávania môže spraviť proces učenia menej monotónnym a viac pútavým.
- **Umožňuje väčšie experimentovanie s rôznymi vyučovacími metódami** – Študenti sa môžu učiť prostredníctvom skúmania, riešenia problémov a skúšania rôznych stratégií. Napríklad programovacie hry, ako Scratch alebo CodeCombat, umožňujú učenie cez prax.
- **Podporuje pozitívne správanie** – Systém odmien a uznaní môže podporiť zodpovednosť a samostatnosť študentov.
- **Zvyšuje socializáciu študentov a vytvára pocit spolupatričnosti** – Spoločné hranie a riešenie úloh podporuje tímovú prácu, komunikáciu a budovanie priateľstiev medzi študentmi.

2.7.2. Nevýhody gamifikácie

Napriek mnohým výhodám môže gamifikácia priniesť aj určité výzvy a riziká, ktoré súvisia s jej implementáciou [16][17]:

- Nie každá téma sa dá gamifikovať.
- Integrácia gamifikácie do existujúcich procesov, kde môže pôsobiť umelo alebo nútene.
- Nadmerná súťaživosť, kde nadmerné sústredenie na súťaž môže viesť k stresu alebo frustrácii.
- Povrchné zapojenie, kde niektorí používatelia môžu byť motivovaní výlučne odmenami, nie skutočným záujmom.
- Vyššie náklady na implementáciu.
- Nedostatok udržateľnosti, kde bez dlhodobých cieľov môže motivácia používateľov časom klesať.

2.8. Terapeutické využitie gamifikácie

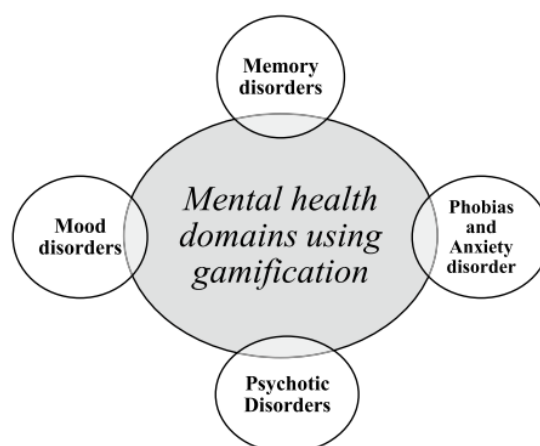
Okrem rôznych vedomostných a edukačných oblastí sa gamifikácia začína presadzovať do rôznych terapeutických oblastí, kde hlavnou úlohou je pomôcť s rôznymi zdravotnými problémami, či už fyzického alebo duševného charakteru.

2.8.1. Gamifikácia pre liečenie duševných porúch

Podľa Svetovej zdravotnej organizácie WHO trpí duševnými problémami ako depresia, úzkosť a psychóza celosvetovo vyše 450 miliónov ľudí. Diagnózy a liečby sa za posledných 10 rokov vyvinuli a dokážu tak lepšie pomôcť pri ich eliminácii. Jednou z nich je gamifikácia. Využitím poznatkov z behaviorálnej ekonómie a klinickej psychológie sa gamifikácia ukazuje ako sľubný nástroj v diagnostike a liečbe duševných porúch. Herné mechanizmy, ako sú odmeny a tresty (napríklad získavanie a strácanie bodov), využívajú princípy operantného podmieňovania, čím motivujú používateľov k opakovanému zapojeniu. Vďaka averzii k strate a tzv. endowment efektu (iracionálnemu ohodnocovaniu vecí, ktoré osoba vlastní resp. nevlastní), môže gamifikácia podporovať pozitívne zmeny v správaní a kognitívnych procesoch, čím sa stáva čoraz populárnejšou medzi odborníkmi a pacientmi [18][19].

Existuje niekoľko modelov (teórií), ktoré gamifikácia môže využívať pri liečbe [18]:

- **Teória klasického a operantného podmieňovania** – Mechanizmy učenia, kde sa správanie formuje na základe asociácií (klasické podmieňovanie) alebo odmien a trestov (operantné podmieňovanie).
- **Teórie očakávania (Expectancy theories)** – Vysvetľujú, ako motivácia závisí od očakávaných výsledkov a vnímania pravdepodobnosti úspechu.
- **Teórie stanovovania cieľov (Goal setting theories)** – Zdôrazňujú význam jasne definovaných cieľov pri motivovaní jednotlivcov k výkonu.
- **Teória sebauvedomenia a autonómie (Self-determination theory, SDT)** – Skúma vnútornú motiváciu a potrebu autonómie, kompetencie a sociálneho prepojenia.
- **Expozitná terapia (Exposure therapy)** – Terapeutická metóda, pri ktorej sa využitím virtuálnej alebo rozšírenej reality jedinca postupne vystavujú podnetom, ktoré vyvolávajú úzkosť, aby sa na nich adaptovali a znížili svoju reakciu.



Obr. 10 - Graf využitia gamifikácie pri psychických problémoch [18]

2.8.2. Gamifikácia pre liečenie fyzických porúch

Využitie herných mechanizmov v rehabilitačnom procese môže výrazne zvýšiť motiváciu pacientov a zlepšiť ich zotavenie. Tradičné terapeutické cvičenia môžu byť pre pacientov monotónne a únavné, čo často vedie k nedostatočnej angažovanosti a nízkej adherencii (súladu) k liečebnému plánu. Gamifikované rehabilitačné programy ponúkajú riešenie tým, že vytvárajú interaktívne a stimulujúce prostredie, ktoré podporuje pravidelnú aktivitu a udržuje pozornosť pacienta [20].

Zvýšenie motivácie a angažovanosti docieľované hraním hier aktivuje dopaminergný systém mozgu, ktorý podporuje učenie a dlhodobé posilnenie nervových spojení. Gamifikácia môže pomôcť udržať pacientov motivovaných k pravidelnému vykonávaniu terapeutických cvičení. Taktiež prepojenie s neurovedou a herným dizajnom umožňuje úspešné nasadenie v terapii, kde sa vyžaduje spolupráca medzi terapeutmi a hernými dizajnérmí. Terapeuti môžu využiť poznatky z herného dizajnu na vytvorenie atraktívnych a efektívnych rehabilitačných programov. Gamifikované terapeutické programy môžu byť prispôbené individuálnym potrebám pacienta, čím sa zabezpečí optimálna úroveň náročnosti a efektivity [20].

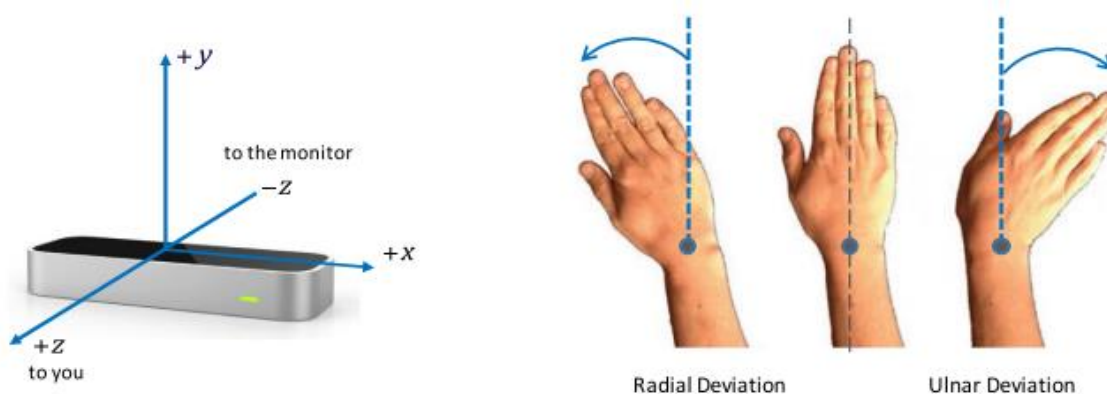
Kombináciou vizuálnych, zvukových a hmatových stimulov v hrách môže pomôcť pacientom lepšie sa sústrediť a urýchliť ich zotavenie. Synchronizované multisenzorické podnety zlepšujú schopnosť mozgu učiť sa nové pohybové vzory. Výhodou hernej spätnej väzby je umožnenie zbierania dát o výkone pacienta, ktoré môžu terapeuti využiť na lepšie prispôbenie terapie a sledovanie progresu v reálnom čase [20].

2.8.3. Gamifikácia pre rehabilitačné nácviky rúk

Okrem využitia gamifikácie pri liečbe mozgových problémov alebo rôznych post traumatických zraneniach je možné využiť gamifikáciu aj pri viac špecializovaných prípadoch. Jedným z nich je využitie pri rehabilitačných cvičeniach nácviku rúk, kde pacienti trpia problémami s nervovou motorikou v oblasti rúk. Vďaka technológiám Leap Motion je možné vytvoriť rehabilitačné cvičenie, kde pomocou zariadenia, ktoré sníma pozíciu ruky resp. rúk v reálnom čase dokáže pacient ovládať lietadlo, ktoré vidí z pohľadu tretej osoby a následne pohybmi rúk mení jeho trajektóriu. Vyhodnocujú sa rôzne pohyby rúk ako uhol vychýlenia ruky od predlaktia (horizontálne alebo vertikálne), vzájomná poloha prstov a taktiež zovretie prstov. Vďaka tomu je možné naprogramovať rôzne funkcie, ktoré budú reagovať na pohyby ruky. Na Obr. 11 môžeme vidieť ako pohyby rúk sú prepojené s vykonávaním určitých úkonov v hre [21].

Hand Gesture							
Therapy	Radial Deviation	Ulnar Deviation	Extension Hyper Extension	Flexion	Abduction/Adduction	Fist Clench	Two Hands with Clenched Fists
Map Movement	Go Left	Go Right	Zoom Out	Zoom In	Move Up	Hold and move the map up, down, left and right	Zoom In, Zoom Out
Normal ROM	0 – 20	0 – 30	0 – 60	0 – 90	0 – 20	Based on movements of Palm, Elbow and Shoulder	Based on distance b/w two Hands
Device	Leap	Leap	Leap	Leap	Leap	Leap	Leap
Body Part	Wrist	Wrist	Wrist, Fingers	Wrist, Fingers	Fingers	Palm, Elbow and Shoulder	Palm, Elbow and Shoulder

Obr. 11 - Reakcia na rôzne pohyby rúk [21]



Obr. 12 - Zariadenie Leap Motion [21]

2.8.4. Gamifikácia a logopédia

V súčasnosti logopédia ako nástroj pomoci pri rečových poruchách sa rozdeľuje do dvoch kategórií podľa veku: detská logopédia a odborná logopédia. Pri detskej logopédii je najdôležitejšie, aby sa pri dieťati odstránila logopedická porucha bez toho, aby si dieťa vyvinulo akýkoľvek psychický blok. Logopéd musí pripraviť prostredie pre dieťa tak, aby čo najmenej pociťovalo, že sa nachádza na vyšetrení alebo cvičení. Aktuálne logopedické cvičenia fungujú tak, že dieťa navštevuje špecialistu v periodických intervaloch a špecialista uskutočňuje s dieťaťom cvičenia podľa typu poruchy. Gamifikácia avšak môže priniesť zásadne zmeny pri pomoci s logopedickými poruchami. Výhodou gamifikácie by bola pomoc nie len pre dieťa, ale aj pre samotného logopéda.

Zo strany dieťaťa sa gamifikácia v logopédii využíva na zvýšenie motivácie a interaktivity pri vzdelávaní a terapii, pričom sa aplikujú herné princípy ako zbieranie bodov, získavanie odmiern či postup na vyšší level. Takéto metódy môžu byť obzvlášť efektívne pri práci s deťmi, keďže podporujú zábavnú formu učenia, zlepšujú koncentráciu a dávajú deťom okamžitú spätnú väzbu. Taktiež sa dokáže eliminovať faktor návštevy ordinácie logopédie a tak ho odbremeniť od potenciálneho

stresu alebo napätia, ktoré môže mať z návštevy lekárov. Gamifikácia v logopédii zahŕňa aktivity, ktoré môžu byť prepojené s mobilnými aplikáciami alebo digitálnymi nástrojmi na precvičovanie výslovnosti a jazykových zručností. Takéto aplikácie často motivujú deti vizuálnymi a zvukovými odmenami, čo zvyšuje ich záujem o terapiu a urýchľuje pokrok. Aktuálny prístup je čoraz viac využívaný nielen v terapii, ale aj vo vzdelávaní všeobecne. Zábavnou formou si vie samotné dieťa dávkovať koľko času bude tráviť nad týmito cvičeniami. Samozrejme je potrebné dodržať určité limity, aby dieťa netrávilo veľa času pri počítači alebo mobilnom zariadení.

Čo sa týka výhod zo strany lekára, obrovskou výhodou je, že dokáže monitorovať výsledky dieťaťa, ktoré môže byť vykonávané zbieraním štatistických údajov ako je napríklad skóre pri každom pokuse alebo čas strávený nad konkrétnym cvičením a podľa toho vyhodnocovať potenciálny progres pri liečbe a upravovať tak cvičenia. Ďalšou výhodou je monitorovanie viacerých pacientov naraz s rôznymi logopedickými problémami. Najčastejšie sa logopedické cvičenia vykonávajú individuálne, ale pri tomto prístupe si vie lekár kedykoľvek pozrieť štatistiky viacerých pacientov v rámci jedného okamžiku a stanoviť tak výsledky pri jednej kontrole. Zároveň to môže slúžiť ako hybridný nástroj, kde dieťa stále môže navštevovať reálne cvičenia, ale zároveň môže vykonávať herné cvičenia, čím lekár môže znížiť návštevnosti dieťaťa pri reálnych cvičeniach.

Gamifikácia pomáha aj dospelým pacientom s logopedickými problémami. Aj keď logopedické poruchy sa v dospelosti (či už nadobudnuté v dospelosti vďaka vonkajším vplyvom alebo nesprávne vyvinutým vyslovovaním) veľmi ťažko liečia. Avšak, moderný trend ukazuje, že mnoho dospelých ľudí hrá počítačové hry, čo pri najmenšom môže pomôcť dospelým pacientom pri zmiernení logopedických porúch a lepším zaradením sa do normálnej spoločnosti.



Obr. 13 - Ukážka hry Poznávaj aplikácie - Logopédia

2.9. Budúcnosť gamifikácie

S aktuálnym trendom vývoja umelej inteligencie a strojového učenia prinesie do gamifikácie značný pokrok. Umelá inteligencia dokáže analyzovať rôzne dáta od rôznych užívateľov omnoho rýchlejšie ako bežný človek. To dáva priestor pre vytvorenie jedinečného zážitku pre každého jednotlivca podľa toho aký štýl alebo zážitok preferuje. Personalizácia úloh alebo odmien tým dokáže omnoho efektívnejšie a objektívnejšie odmeňovať ľudí bez toho, aby stratili motiváciu pre dosiahnutie svojho cieľa. V súčasnosti sa gamifikácia nachádza aj v aplikáciách, ktoré nemajú nič spoločné s hrami, ale využívajú aspekty ako úlohy typu: „vykonaj túto činnosť a my ťa odmeníme týmto“. Medzi takéto aplikácie môžeme zaradiť aj bežné aplikácie nákupných centier, ktoré nasadili trend získavania bodov pri nakupovaní a za odmenu dokážu zákazníci nákupných centier využívať získané body na rôzne kupóny podľa toho čo preferujú [22].

Ďalším aspektom, ktorý dokáže priniesť inováciu do gamifikácie je virtuálna a rozšírená realita (VR/AR). Pri virtuálnej realite užívateľ vstupuje priamo do virtuálneho sveta podľa naprogramovanej aplikácie. To môže slúžiť ako simulačná pomôcka pri napríklad medicínskych tréningoch ako napríklad štúdium chirurgie, kde užívateľ sa snaží previesť operáciu na virtuálnom pacientovi a podľa toho ako prevedie operáciu tak bude ohodnotený určitým počtom bodov. Taktiež dokáže pomôcť pri zaškoľovaní pilotov štandardných dopravných lietadiel alebo vojenských lietadiel pomocou simulátorov, kde sa dokážu vžiť do danej situácie a následne v reálnej situácii využiť znalosti nadobudnuté virtuálnym tréningom [22].

Čo sa týka rozšírenej reality, tá oproti virtuálnej realite poskytuje rozšírenie vnímanie reality zobrazovaním rôznych detailov priamo užívateľovi. To môže priblížiť užívateľa pri vykonávaní istého úkonu a hneď mu ukázať ako bol jeho výsledok prijatý realitou. Vzhľadom na neustály rozvoj technológií sa očakáva, že gamifikácia bude v budúcnosti zohrávať ešte významnejšiu úlohu vo vzdelávaní a jej aplikácie budú stále sofistikovanejšie [22].



Obr. 14 - Budúcnosť gamifikácie pomocou VR [22]

3. Návrh logopedickej platformy LogoHra

V tretej kapitole diplomovej práce podrobne ukážeme vytvorenie logopedickej webovej platformy LogoHra, ktorá je finálnym výstupom diplomovej práce.

3.1. Cieľ práce

Cieľom práce bolo vytvorenie webovej platformy, ktorej úlohou je pomôcť deťom v predškolskom a školskom veku s logopedickými problémami. Hlavnou vlastnosťou bolo vytvoriť platformu, ktorú je možné upravovať, nastavovať a vytvárať rôzne typy úloh podľa požiadaviek logopedického odborníka. Platforma mala pozostávať z dvoch hlavných sekcií. Prvou je samotná hra, ktorá funguje na princípe pohybu na hernom poli pomocou hodu kockou a zodpovedaním vygenerovanej otázky zo zoznamu. Druhou sekciou je grafické administrátorské rozhranie, ktoré umožňuje administrátorovi pridávať, upravovať a priradzovať úlohy a hry a tým docieľiť rôznorodosť hier. Po vytvorení oboch sekcií a ich nasadení na server má vzniknúť funkčná platforma, ktorú je možné nasadiť do reálnej prevádzky.

3.2. Použité technológie

Platforma sa skladá z niekoľkých technológií a jazykov, ktoré majú za úlohu vykonávať rozličné funkcie, kde ich prepojenie je kľúčové pre správne fungovanie platformy.

3.2.1. HyperText Markup Language

HyperText Markup Language (HTML) je tagovací jazyk používaný na tvorbu webových stránok. Definuje štruktúru webového obsahu pomocou značiek (tagov), ktoré obklopujú rôzne prvky, ako sú texty, obrázky, odkazy či formuláre. V súčasnosti nie je použiteľný samostatne pretože mu chýbajú mnohé funkcie ako úprava dizajnu, dynamické efekty, kde je potrebné vytvoriť spojenie s inými technológiami. Aktuálnou verziou jazyka je HTML5, ktorá umožňuje lepšiu podporu multimédií, interaktívnych prvkov a zlepšuje výkon webových stránok. Pracuje s príponou .html.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
<ul>
  <li><a href="#">Domov</a></li>
  <li><a href="#">Kontakt</a></li>
</ul>
<h2>Nový článok</h2>
<p>Toto je obsah článku.</p>
</body>
</html>
```

Obr. 15 - Ukážka štruktúry HTML jazyka

3.2.2. Cascading Style Sheets

Cascading Style Sheets (CSS) patrí do kategórie opisných tagovacích jazykov, ktorého hlavnou funkciou je upravovať dizajn webovej stránky. Medzi dizajnové zmeny patrí menenie farieb textu, zmena štýlu písma, rozloženie elementov, zmena pozadia, úprava veľkosti objektov, obrázkov a pod. Je zodpovedný za responzivnosť stránky, čo znamená že stránka sa bude meniť podľa veľkosti displeja (hlavne pre mobilné zariadenia). CSS vieme rozdeliť na inline, interné a externé, avšak nevie pracovať samostatne a je ho potrebné prepojiť z HTML. V súčasnosti je aktuálna verzia CSS3. Pre CSS existuje mnoho šablón (frameworkov), ktorých úlohou je zjednodušiť prácu s dizajnom. Medzi najznámejšie šablóny patrí Bootstrap, Tailwind, Foundation atď.

```
.game-button {  
  background: #007bff;  
  color: white;  
  text-decoration: none;  
  padding: 8px 12px;  
  border-radius: 5px;  
  transition: 0.3s;  
}
```

Obr. 16 - Ukážka štruktúry CSS jazyka

3.2.3. Bootstrap

Bootstrap je populárny voľne dostupný frontend framework, ktorý sa používa na rýchly a jednoduchý vývoj responzivných a mobilne priateľských webových stránok a aplikácií. Bol vytvorený spoločnosťou Twitter a prvýkrát bol vydaný v roku 2011. Bootstrap poskytuje množstvo preddefinovaných HTML, CSS a JavaScript komponentov, ktoré môžu byť ľahko použité na vytvorenie moderných a vizuálne atraktívnych webových rozhraní. Tvorca webových aplikácií tak nemusí písať pre každý blok rôzne CSS vlastnosti, ale môže použiť predefinované triedy, ktoré sú automaticky prednastavené a umožnia tak rýchle a prehľadnejšie nastavovanie dizajnu.

```
<!-- ✅ FOOTER -->  
<footer class="p-3 bg-black text-light">  
  <div class="container d-flex align-items-center justify-content-between me-0">  
    <span class="ms-auto">@ <%= date( format: "Y") %> Platforma LogoHra. KEMT FEI TUKE. Všet  
  
    <div class="d-flex align-items-center gap-5 ms-5">  
      <span>  
        📞 <a href="tel:+421556022334" class="text-light text-decoration-none">+421 5  
        ✉️ <a href="mailto:kemt.fei@tuke.sk" class="text-light text-decoration-none">  
      </span>  
    </div>  
  </div>
```

Obr. 17 - Ukážka využitia frameworku Bootstrap v HTML kóde

3.2.4. JavaScript

JavaScript (JS) je programovací jazyk, ktorý umožňuje vytvárať interaktívne prvky na webových stránkach. Na rozdiel od HTML a CSS, ktoré sa starajú o štruktúru a vzhľad stránky, JavaScript riadi jej správanie. Pomocou neho je možné dynamicky meniť obsah stránky, upravovať existujúce HTML prvky, meniť CSS vlastnosti alebo komunikovať so serverom cez backend technológie ako PHP.

Dnešné webové aplikácie často využívajú buď čistý JavaScript, alebo moderné frameworky, ktoré uľahčujú prácu vývojárom vďaka predpripraveným knižniciam a funkciám. Medzi najpopulárnejšie JS frameworky patria React.js, Angular.js a Vue.js, ale aj nástroje ako Bootstrap, ktoré obsahujú rôzne interaktívne komponenty. Je možné taktiež využiť rôzne špecializované frameworky ako napríklad herný framework Phaser 3.

```
// Úprava kategórie
$(document).on("click", ".edit-btn", function () :void {
    let categoryId = $(this).data("id");
    let categoryItem = $('#category-${categoryId}');
    let inputField = categoryItem.find(".edit-category-name");

    inputField.show().focus(); // Zobrazíme pole na úpravu a na
    categoryItem.find(".save-btn").show();
    $(this).hide();
});
```

Obr. 18 - Ukážka štruktúry JS jazyka

3.2.5. Phaser 3

Phaser 3 je framework určený pre vytváranie 2D hier a interaktívneho obsahu pre webové prehliadače. Je napísaný v JavaScripte a je vhodný pre vývoj hier, ktoré bežia na rôznych platformách, vrátane desktopových a mobilných zariadení. Phaser 3 sa skladá z troch hlavných funkcií, ktoré slúžia na inicializáciu a načítanie rôznych elementov, obrázkov a zvukových nahrávok, vytvorenie týchto elementov a následnú aktualizáciu pri zmene. Jeho využitím je možno vytvoriť rôzne 2D hry jednoduchšie než s využitím klasického JavaScriptu.

```
// Inicializácia Phaseru
const config = {
    type: Phaser.AUTO,
    scale: {
        mode: Phaser.Scale.RESIZE,
        parent: 'game-container', // Nastaví rodičovský element pre hru (môže byť ID elementu alebo samotný element)
        autoCenter: Phaser.Scale.CENTER_BOTH // Centrovat hru
    },
    scene: {
        preload: preload,
        create: create,
        update: update,
    }
}
```

Obr. 19 - Inicializácia Phaser 3 v kóde

3.2.6. jQuery a AJAX

jQuery je JavaScript knižnica, ktorá uľahčuje manipuláciu s HTML dokumentom, správu udalostí, animácie a AJAX požiadavky. Umožňuje jednoduchšiu a efektívnejšiu prácu s DOM (Document Object Model) a je kompatibilná s rôznymi prehliadačmi. Špecifickou vlastnosťou jQuery príkazov je začiatkový symbol „\$“, ktorý hovorí o použití knižnice jQuery.

```
$("#save-dataset").on("click", function () :void {  
    let datasetName = $("#dataset-name").val();  
    if (datasetName === "") {  
        alert("Názov datasetu nemôže byť prázdny!");  
        return;  
    }  
})
```

Obr. 20 - Ukážka využitia knižnice jQuery

AJAX (Asynchronous JavaScript and XML) je technológia umožňujúca asynchrónnu komunikáciu medzi klientom (prehliadačom) a serverom bez nutnosti znovu načítať celú stránku. Aj keď názov obsahuje "XML", dnes sa častejšie používajú formáty ako JSON.

```
// AJAX call to manage games  
$.post("manage_games_ajax.php", {action: "get_game", game_id: gameId}, function (response) :void {  
    let games = JSON.parse(response);  
})
```

Obr. 21 - Ukážka volania AJAX pre komunikáciu so serverom

3.2.7. Web Speech API

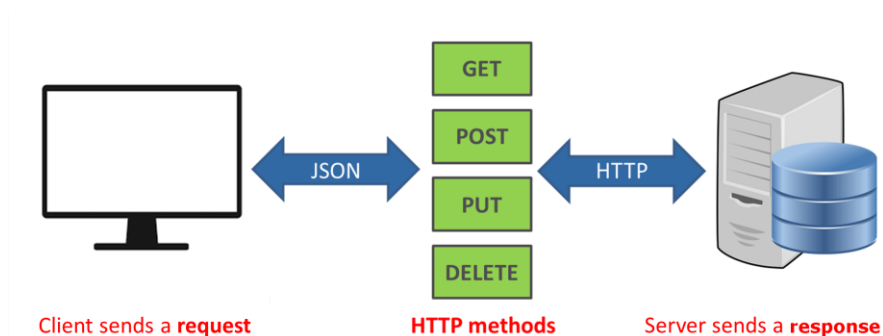
Web Speech API je špeciálna knižnica od spoločnosti Google, ktorá umožňuje rozpoznávať ľudskú reč a následne reagovať s webovou aplikáciou. Implementácia sa vykonáva pomocou jazyka JavaScript. Aktuálne existujú dve verzie knižnice: Web Speech API a Speech-to-Text.

Web Speech API je pôvodná verzia rozpoznávania reči, ktorá je dostupná zadarmo a funguje na všetkých webových prehliadačoch, ktoré disponujú jadrom Chromium. Má isté obmedzenia ako napríklad odstránenie miery dôveryhodnosti vysloveného slova pre slovenský jazyk, ale je to stále kvalitná a spoľahlivá verzia.

Speech-to-Text je vylepšená verzia, ktorá pracuje na službách Google Cloud (taktiež nazývaná Speech-to-Text V2). Hlavnou výhodou je efektívnejšie a kvalitnejšie rozpoznávanie reči. Hlavnou nevýhodou je obmedzenie, pretože táto verzia je už spoplatnená a aj na bezplatné využitie je potrebné zadať kreditné údaje, čo môže niektorých vývojárov demotivovať, ak ide o experimentálny vývoj.

3.2.8. Hypertext Preprocessor

PHP (Hypertext Preprocessor) je skriptovací jazyk, ktorý sa vykonáva na strane servera. Na rozdiel od HTML, CSS a JavaScriptu, ktoré bežia na strane klienta a nevyžadujú server, PHP potrebuje serverové prostredie na svoje fungovanie. Hlavnou úlohou je spracovanie dát a ich následné odoslanie klientovi. PHP je tiež výborným nástrojom na prepojenie webových stránok s databázami a manipuláciu s údajmi. Kód môže byť uložený v samostatných súboroch s príponou .php alebo vložený priamo do HTML súboru. V takom prípade je potrebné zmeniť príponu z .html na .php. Dokáže tiež komunikovať s JavaScript časťou aplikácie, čím rozširuje možnosti dynamického obsahu.



Obr. 22 - Princíp fungovania komunikácie klient-server

3.2.9. MySQL

MySQL je relačný databázový systém (RDBMS), ktorý umožňuje efektívne ukladanie, spravovanie a manipuláciu s dátami pomocou jazyka SQL (Structured Query Language). Dáta sú organizované v tabuľkách, ktoré môžu byť vzájomne prepojené prostredníctvom relácií, čím sa minimalizuje redundancia informácií. Každá tabuľka pozostáva zo stĺpcov a riadkov – stĺpce definujú typ údajov, ktoré sa v nich uchovávajú, určujú ich dátový typ a môžu obsahovať rôzne obmedzenia, napríklad požiadavku na unikátne hodnoty.

Riadky následne predstavujú konkrétne uložené dáta. Na prácu s tabuľkami sa využívajú SQL dopyty, ktoré umožňujú manipuláciu a načítanie údajov. MySQL patrí medzi najrozšírenejšie databázové systémy pre webové aplikácie a dá sa jednoducho integrovať s rôznymi programovacími jazykmi, ako napríklad PHP.

```
CREATE TABLE user_games (
  user_games_id INT UNSIGNED PRIMARY KEY AUTO_INCREMENT,
  user_id INT UNSIGNED NOT NULL,
  game_id INT UNSIGNED NOT NULL,
  FOREIGN KEY (user_id) REFERENCES users(id) ON UPDATE CASCADE ON DELETE CASCADE,
  FOREIGN KEY (game_id) REFERENCES games(id) ON UPDATE CASCADE ON DELETE CASCADE
);
```

Obr. 23 - Ukážka MySQL príkazov

3.2.10. XAMPP

Pre vytvorenie správnej komunikácie klient-server je potrebné nainštalovať softvér, ktorý umožní vytvorenie servera pre sieťovú komunikáciu a databázu. XAMPP je voľne dostupný program, ktorý obsahuje niekoľko serverových služieb, kde pri ich spustení vieme vytvoriť Apache a MySQL server, potrebný pre zrealizovanie vytvorenia platformy.

Výhodou programu je jednoduchosť, kde stačí program nainštalovať, zapnúť potrebné služby a vložiť webovú aplikáciu do priečinka htdocs umiestneného v koreňovom adresári C:/xampp. Vďaka tomu vieme simulovať reálnu prevádzku na jednom počítači a pripraviť funkčnú webovú aplikáciu. Komunikácia následne prebieha na IP adrese 127.0.0.1 (localhost) v rámci počítača alebo na IP adrese z rozsahu lokálnej siete, kde je počítač pripojený.



Obr. 24 - Ukážka aplikácie XAMPP

3.3. Štruktúra platformy

Na Obr. 25 môžeme vidieť štruktúrne rozloženie platformy. Platforma sa delí na dve základné časti: používateľskú a administrátorskú. Používateľská časť sa rozdeľuje podľa toho, či používateľ je prihlásený. Ak používateľ nie je prihlásený, tak je mu umožnený prístup iba ku úvodnej stránke a samotnej hre. Prihlásený používateľ má okrem toho prístup aj ku špecifickým hrám, ktoré mu môže prideliť administrátor. Zároveň má prístup k možnosti zobrazenia svojho používateľského konta, v ktorom môže vidieť svoj profil a taktiež absolvované hry.

Administrátor má všetky vyššie uvedené právomoci, ale zároveň má prístup aj ku funkciám, ku ktorým obyčajný používateľ nemá prístup. Administrátor môže upravovať celú štruktúru ako bude hra vyzerať a aké úlohy a obrázky bude načítavať. Administrátorské rozhranie je rozdelené do 6 častí:

- Správa hier,
- Správa datasetov,
- Správa úloh,
- Správa obrázkov,
- Správa kategórií obrázkov,
- Správa používateľov.

V každej z kategórií je možné upraviť a vytvoriť prepojenie pre správne fungovanie platformy. Správa hier umožňuje vytvárať, upravovať a spravovať hry, ktoré následne môžu užívatelia vidieť. Správa datasetov má za úlohu spájať rôzne úlohy a obrázky to jednej skupiny, ktorú je následne možné pridať do hry. Správa úloh, obrázkov a ich kategórií má za úlohu spravovať pridávanie elementárnych častí, ktoré sa budú môcť pridať vo vyššie uvedených správach. Správa používateľov umožňuje kontrolu používateľov a ich prehľad o výsledkoch v hrách. Administrátor vidí výsledky všetkých hráčov, ktorí sú zaregistrovaní v platforme.



Obr. 25 - Štruktúra platformy

3.4. Návrh a dizajn platformy

V podkapitole 3.4 detailne povieme čo bude každá podstránka obsahovať, akú funkciu bude plniť a aký bude mať dizajn. Administrátorská časť pozostáva z formulárov a rôznych zobrazení preto je dôležité, aby pre administrátora, ktorý môže byť laik v oblasti informačných technológií sa vedel navigovať a aby toto prostredie bolo prehľadné. Čo sa týka hernej časti, tá musí byť prispôbená pre detských používateľov, teda musí obsahovať viac graficky prijateľné prostredie (obrázky, farby, rôzne animácie) a minimum zložitého textu.

3.4.1. Návrh a dizajn navigačnej lišty

Na začiatok je potrebné navrhnuť element, ktorý je zdieľaný pre celú platformu a mení sa dynamicky. Navigačná lišta na pravej strane obsahuje funkcionality zaregistrovania nového používateľa a prihlásenie zaregistrovaných používateľov. Pri obidvoch možnostiach sa pri kliknutí na tlačidlo zobrazí formulár, ktorého vyplnením sa vykoná požadovaný úkon. Po úspešnom zaregistrovaní alebo prihlásení je používateľ prihlásený a tlačidlá zmenia svoju funkciu. Tlačidlá následne umožňujú funkciu odhlásenia alebo zobrazenia detailov používateľského účtu, prípadne administrátorskú časť, ak je prihlásený administrátor. Na ľavej strane navigačnej lišty sa zobrazuje meno prihláseného používateľa alebo správa, že používateľ nie je prihlásený.

Dizajn navigačnej lišty pozostáva zo štyroch blokov, ktoré menia svoj obsah. Prvým blokom je výpis textu, ktorý zobrazuje meno používateľa. Na pravej strane obsahuje 2 až 4 tlačidlá, ktoré sa menia podľa funkcionality. Tlačidlá sú farebne a tvarovo odlišné podľa typu funkcie. Registračný formulár postáva zo siedmich vstupných blokov, do ktorých je možné napísať hodnoty. Taktiež obsahuje možnosť pridať obrázok a jeho zobrazenie. V pravom dolnom rohu je potvrdzovacie tlačidlo. Prihlasovací formulár je podobný so zredukovaným počtom vstupných blokov. Na Obr. 26 môžeme vidieť grafický dizajn.

The diagram illustrates the design of the navigation bar and two forms: 'Registrácia nového používateľa' (New User Registration) and 'Prihlásenie' (Login).

Navigation Bar: A horizontal bar with a grey background. On the left, there is a box labeled 'MENO PRIHLÁSENÉHO'. On the right, there are three buttons labeled 'TLAČIDLO'.

Registrácia nového používateľa: A form with a title bar and a close button. It contains a profile picture placeholder labeled 'PROFIL OBR.' with a circular icon 'OBR'. The form fields include: 'Používateľské meno' (Username), 'Rok narodenia' (Year of birth) with a dropdown menu showing '20XX', 'Meno' (First name), 'Priezvisko' (Surname), 'E-mail', and 'Heslo' (Password). There is a checkbox labeled 'Vytvoriť admin účet?' (Create admin account?) and a blue button labeled 'VYTVORIŤ' (Create).

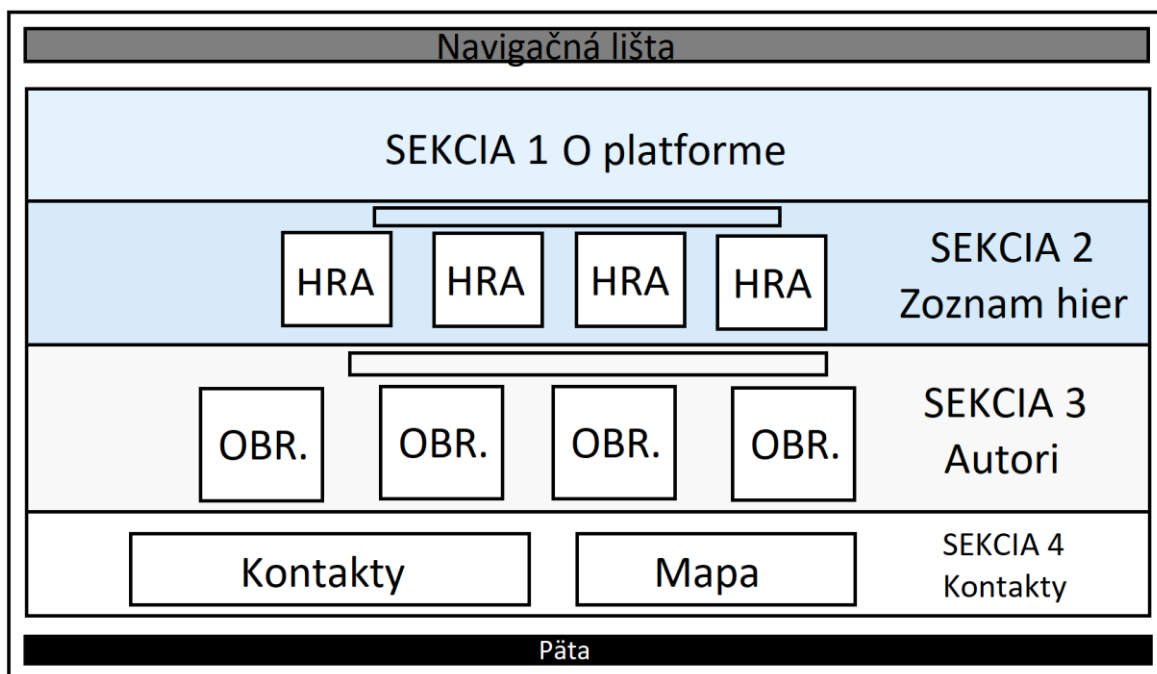
Prihlásenie: A form with a title bar and a close button. It contains a single input field for 'Používateľské meno alebo e-mail' (Username or email) and a 'Heslo' (Password) field. There is a checkbox labeled 'Ostať prihlásený' (Stay logged in) and a blue button labeled 'PRIHLÁSIŤ SA' (Log in).

Obr. 26 - Návrh navigačnej lišty a formulárov na prihlásenie/registáciu

3.4.2. Návrh a dizajn úvodnej podstránky

Úvodná podstránka, taktiež označovaná ako index, plní hlavne prezentačnú časť platformy. Obsahuje informácie o platforme, autoroch a kontaktné údaje. Hlavnou časťou je podstránky je zobrazenie aktuálnych hier, ktoré sa zobrazujú podľa toho aké sú vytvorené. Zároveň, ak je používateľ prihlásený môžu sa zobrazíť hry, ktoré mu administrátor prideliť a nie sú verejne dostupné. Súčasťou je aj odkaz na dokument, ktorý umožní novému administrátorovi zoznámenie sa s platformou a ako vytvárať nové hry.

Na Obr. 27 môžeme vidieť dizajn podstránky. Skladá sa z troch veľkých skupín. Prvou skupinou je navigačná lišta, ktorej návrh a dizajn sme si vysvetlili v odseku 3.4.1. Druhou skupinou je obsah podstránky, ktorá je rozdelená do štyroch sekcií. Prvá sekcia obsahuje textové polia informujúce používateľa. Druhá sekcia obsahuje bloky s hrami, ktoré sú dynamicky generované podľa zoznamu z databázy. Tretia sekcia obsahuje informácie o tvorcach platformy a taktiež štyri obrázky zobrazujúce vedúcich a autora práce. Poslednou sekciou sú kontakty, ktorá obsahuje na časti: kontaktnú (obsahujúca telefónne číslo, e-mailovú adresu a adresu) a grafickú, ktorá je reprezentovaná mapou s presnou lokalitou. Na záver podstránky je umiestnená päta, ktorá obsahuje autorské práva a taktiež kontaktné údaje. Táto päta spoločne s navigačnou lištou bude použitá pre každú podstránku v nasledujúcich podkapitolách.

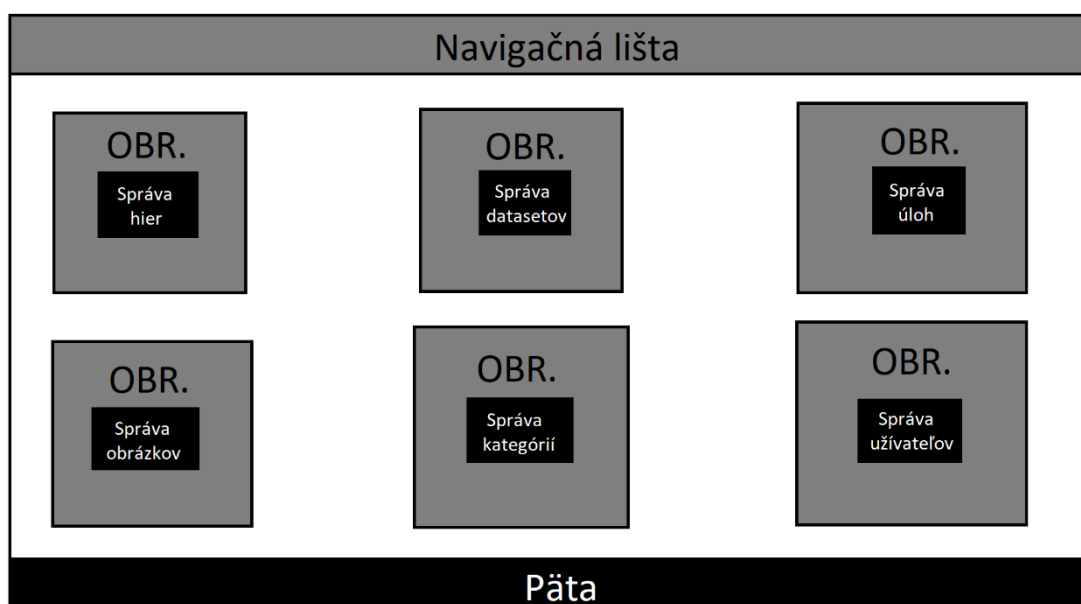


Obr. 27 - Dizajn úvodnej podstránky

3.4.3. Návrh a dizajn administrátorského rozhrania

Rozhranie administrátora je prepojovacia podstránka, ktorá umožňuje prechod na podstránky s rôznou funkcionalitou. Po kliknutí na zvolenú možnosť je užívateľ automaticky presmerovaný na požadovanú podstránku.

Dizajn pozostáva zo šiestich veľkých blokov, ktoré sú zložené z obrázkov indikujúcich požadované funkcionality. Obrázky dopĺňujú nadpisy, ktoré jednoznačne označujú názov funkcionality. Pri prechode kurzorom na konkrétny blok sa daný blok zväčší, aby sa jednoznačne dalo určiť, kde sa aktuálne kurzor nachádza. Zároveň sú bloky aktívne odkazy umožňujúce presmerovanie.



Obr. 28 - Dizajn administrátorského rozhrania

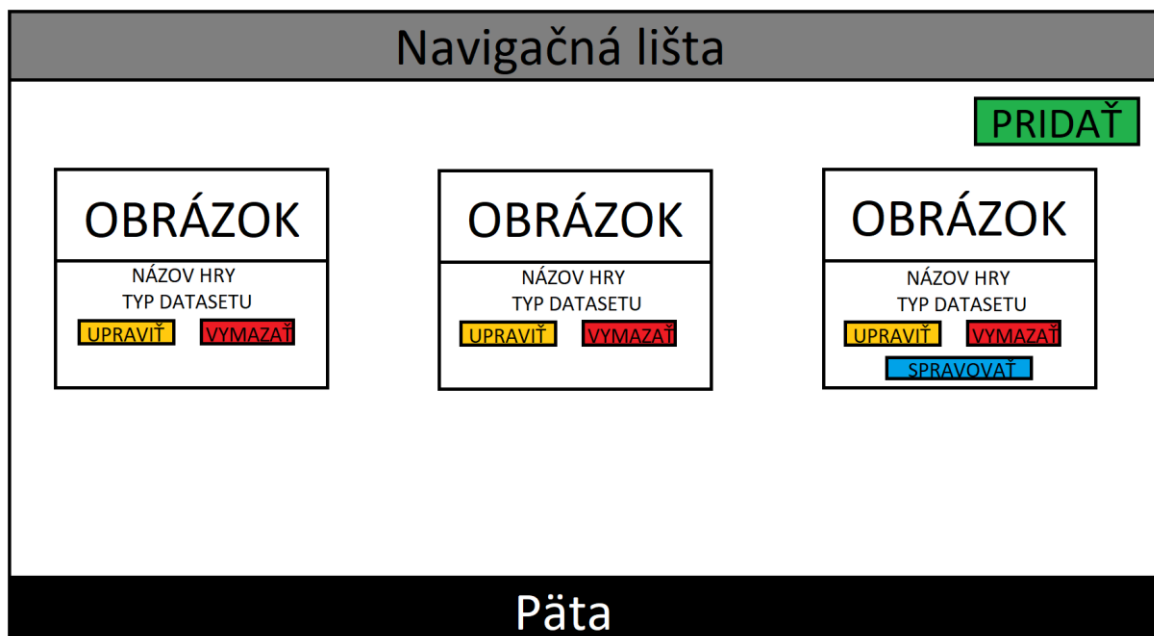
3.4.4. Návrh a dizajn podstránky správy hier

Ako bolo vysvetľované v odseku 3.3, podstránka správy hier umožňuje vytváranie hier. Pridávanie je vykonávané pomocou tlačidla, ktoré aktivuje formulár, do ktorého bude možné zadať vstupné hodnoty ako názov, precvičovaný dataset, obrázky pre pozadie, textúry hráčov a mapu, ktoré sa budú vykresľovať v hre a či daná hra má byť verejná alebo súkromná.

Základnou vlastnosťou je výpis vytvorených hier. Po vytvorení sa zobrazí hra v bloku, kde je obrázok, názov hry a použitý dataset. Následne môže administrátor vykonávať zmeny ako úpravu hry (názov, dataset, obrázky, ...), odstránenie hry alebo ak je hra neverejná tak možnosť spravovať prístup ku danej hre.

Obr. 29 ilustruje zobrazovanie vytvorených hier. Hry sa zobrazujú do blokov, ktoré sú rozdelené do troch častí. V prvej časti je znázornený obrázok hry, v druhej sú textové bloky oznamujúce názov

a hry a typ datasetu. V poslednej časti sú farebné tlačidlá slúžiace na upravenie (žlté tlačidlo), odstránenie (červené tlačidlo) a v niektorých prípadoch na spravovanie prístupu k hre (modré tlačidlo). Na pravej strane je zelené tlačidlo, ktoré pri kliknutí zobrazí formulár pozostávajúci zo vstupného bloku na názov hry, výber datasetu a štyroch vstupných blokov, ktoré prijímajú iba súbory.



Obr. 29 - Dizajn podstránky správy hier

3.4.5. Návrh a dizajn podstránky správy datasetov

Správa datasetov má za úlohu spájať jednotlivé úlohy a obrázky do jednej skupiny, ktorá sa môže následne trénovať. Podstránka umožňuje vytvorenie nového datasetu a jeho odstránenie, ak sa daný dataset už nepoužíva. Následne pri výbere datasetu je možné priradiť úlohy (jednoduché slová alebo hádanky) a obrázky podľa toho, čo chce administrátor testovať. Pri výbere sa vyberá zo zoznamu aktuálnych slov databáze. Možnosť pridať úlohu je obmedzená na úlohy, ktoré ešte nie sú v datasete, kde jedna úloha môže byť priradená do niekoľkých rôznych datasetov. Následne po pridaní môže úlohy vymazať podľa potreby. Filter umožňuje jednoduchšie vyhľadávanie podľa typu úlohy.

Dizajn podstránky sa rozdeľuje na dve veľké časti. Prvou časťou je blok, ktorý obsahuje zoznam datasetov usporiadaných do vertikálne umiestnených blokov. Tlačidlo „+ Dataset“ umožní vytvorenie nového datasetu a je umiestnené nad zoznamom datasetov. Napravo sa nachádza výpis datasetu rozdelený do troch blokov podľa toho o aký typ úlohy sa jedná. Nad týmto výpisom sa nachádzajú 2 tlačidlá umožňujúce vykonať úpravy datasetu rozlíšené rôznymi farbami. Pri kliknutí na filter sa zobrazí zoznam možnosti podľa typu kategórie úlohy. Obr. 30 ilustruje dizajn podstránky.

Obr. 30 - Dizajn podstránky správy datasetov

3.4.6. Návrh a dizajn podstránky správy úloh

Správa úloh zabezpečuje pridávanie a úpravu slovných úloh, ktoré sa budú využívať v samotnej hre. Je možné spravovať dva typy úloh: jednoduché slová a hádanky. Oba typy sa pridávajú pomocou formulára, ktorý následne uloží dané úlohy do databázy. Hádanky obsahujú dva vstupné hodnoty: samotné znenie hádanky a odpoveď na ňu. Jednoduché slová obsahujú jedno slovo. Všetky pridané úlohy sú následne zaznamenané a vypísané pod formulármi na pridávanie. Každá úloha je modifikovateľná pomocou tlačidiel, ktoré podľa typu úloh zobrazia priamo vo výpise vstupné polia s vopred vyplnenými hodnotami. Odstránenie je umožnené pomocou druhého tlačidla.

Dizajn podstránky spočíva v dvoch veľkých sekciách, ktoré sú navzájom oddelené. V prvej sekcii je riešená správa hádaniek. Správa hádaniek je rozdelená na dve časti. Prvá časť obsahuje blok pre formulár s dvoma vstupnými poľami a modrým potvrdzovacím tlačidlom pod formulárom. Druhá časť vypisuje zoznam hádaniek s odpoveďami pod sebou. Pri každej hádanke je na pravej strane umiestnené žlté tlačidlo na úpravu a červené tlačidlo na odstránenie. Druhou sekciou je správa jednoduchých slov, ktorá je umiestnená pod sekciou hádaniek. Obsahuje zjednodušený formulár s jedným vstupným poľom a modrým tlačidlom na potvrdenie. Výpis jednotlivých slov je realizovaný po malých blokoch umiestnených vedľa seba s vypísaným slovom a dvoma malými tlačidlami pre úpravu a odstránenie s dôvodu zbytočného využitia priestoru. Tento spôsob umožní lepšiu reakciu podstránky na rôznu šírku obrazovky. Na Obr. 31 môžeme vidieť dizajn podstránky pre správu úloh.

Obr. 31 - Dizajn podstránky pre správu úloh

3.4.7. Návrh a dizajn podstránky správy obrázkov

Podstránka patrí medzi jednu z najkomplexnejších podstránok na zobrazenie. Obsahuje zoznam všetkých obrázkov, ktoré sú uložené v databáze. Každý obrázok má pridelený názov a kategóriu, pod ktorými bol vytvorený a taktiež možnosť nahratia nového obrázka. Obrázky majú priradené vlastné tlačidlá pre potvrdenie úpravy a odstránenie. Na podstránke je umiestnený aj formulár, ktorý plní úlohu pridávania nových obrázkov do databázy. Ten umožňuje vložiť názov obrázka, do akej kategórie patri a samotný obrázok v obrázkovom formáte.

Dizajn podstránky znázornený na Obr. 32 zobrazuje formulár, ktorý je umiestnený v hornej časti podstránky a zoznam obrázkov umiestnený na dolnej časti podstránky. Formulár obsahuje tri vstupné polia, ktoré sú umiestnené pod sebou a sú zakončené modrým potvrdzovacím tlačidlom. Zoznam obrázkov pridávaných do databázy sa skladá z niekoľkých blokov usporiadaných do riadkov, ktorých veľkosť a počet je responzívny od šírky stránky. Každý takýto blok má vertikálne zoradené elementy pre samotný obrázok, názov, kategóriu a možnosť nahráť nový obrázok. Na konci bloku sa nachádza žlté tlačidlo pre potvrdenie úpravy obrázku a červené tlačidlo pre odstránenie obrázka zo zoznamu.

The design of the image management page is structured as follows:

- Navigation Bar (Navigačná lišta):** A grey bar at the top of the page.
- Form (FORMULÁR):** A central white box containing:
 - Input fields for "Názov obrázka" (Image Name), "Kategória obrázka" (Image Category), and "Súbor" (File).
 - A text input field with the placeholder "image_name.png".
 - A blue button labeled "PRIDAŤ" (Add).
- Image List:** A row of image cards below the form. Each card is labeled "OBR" and contains:
 - A placeholder image.
 - A text input field with the placeholder "image_name.png".
 - A yellow button labeled "UPRAVIŤ" (Edit).
 - A red button labeled "VYMAZAŤ" (Delete).
- Footer (Päta):** A black bar at the bottom of the page.

Obr. 32 - Dizajn podstránky správy obrázkov

3.4.8. Návrh a dizajn podstránky správy kategórií obrázkov

Správa kategórií obrázkov je veľmi jednoduchá podstránka, kde hlavnou a jedinou funkciou je vytvoriť, upravovať a vymazávať názvy kategórií obrázkov, ktoré sú neskôr pridelované samotným obrázkom ako bolo navrhované v odseku 3.4.7. Dôvodom tohoto návrhu je eliminácia duplicitných názvov a chýb, ktoré by mohli nastať pri ručnom písaní názvov kategórií pri pridávaní nových obrázkov.

Na Obr. 33 môžeme vidieť dizajn podstránky. Skladá sa z jednoduchého bloku v hornej časti podstránky. Kde obsahuje jedno vstupné pole a modré potvrdzovacie tlačidlo na pravej strane. Pod formulárom sa nachádza blok vypísaných kategórií obrázkov zoradených do riadkov a stĺpcov. V každom bloku kategórie sa nachádza textová časť s názvom kategórie a modrým oznamovacím blokom, ktorý indikuje počet obrázkov, ktoré využívajú konkrétnu kategóriu. Pod textovými výpismi sa nachádzajú dve tlačidlá pre úpravu názvu kategórie alebo jej odstránenie.

Navigačná lišta

Názov kategórie PRIDAŤ

NÁZOV POČET
UPRAVIŤ VYMAZAŤ

NÁZOV POČET
UPRAVIŤ VYMAZAŤ

NÁZOV POČET
UPRAVIŤ VYMAZAŤ

NÁZOV POČET
UPRAVIŤ VYMAZAŤ

NÁZOV POČET
UPRAVIŤ VYMAZAŤ

NÁZOV POČET
UPRAVIŤ VYMAZAŤ

Päta

Obr. 33 - Dizajn podstránky správy kategórie obrázkov

3.4.9. Návrh a dizajn podstránky zoznamu používateľov

Podstránka plní jednoduchú funkciu výpisu všetkých zaregistrovaných používateľov. Pre bežného užívateľa nie je dostupná možnosť vidieť iných používateľov, preto je podstránka potrebná pre administrátorov, ktorí si chcú pozrieť výsledky svojich používateľov. Administrátor má taktiež dostupnú funkciu vyhľadávania používateľov podľa ich prihlasovacieho mena. Administrátor môže taktiež odstrániť užívateľov, ale nemôže odstrániť iných administrátorov.

Dizajn pozostáva so vstupného poľa „HĽADAŤ“, ktoré je umiestnené nad zoznamom. Zoznam tvoria bloky, v ktorých sú pod sebou zapísané údaje o používateľovi. Pod textovou časťou sa nachádza modré tlačidlo, ktoré presmeruje na konkrétneho používateľa a červené tlačidlo, ktoré umožní jeho vymazanie (pri iných administrátoroch sa táto možnosť nezobrazí).

Navigačná lišta

HĽADAŤ

ÚDAJE
ZOBRAZIŤ VYMAZAŤ

ÚDAJE
ZOBRAZIŤ VYMAZAŤ

ÚDAJE
ZOBRAZIŤ VYMAZAŤ

ÚDAJE
ZOBRAZIŤ VYMAZAŤ

ÚDAJE
ZOBRAZIŤ VYMAZAŤ

ÚDAJE
ZOBRAZIŤ VYMAZAŤ

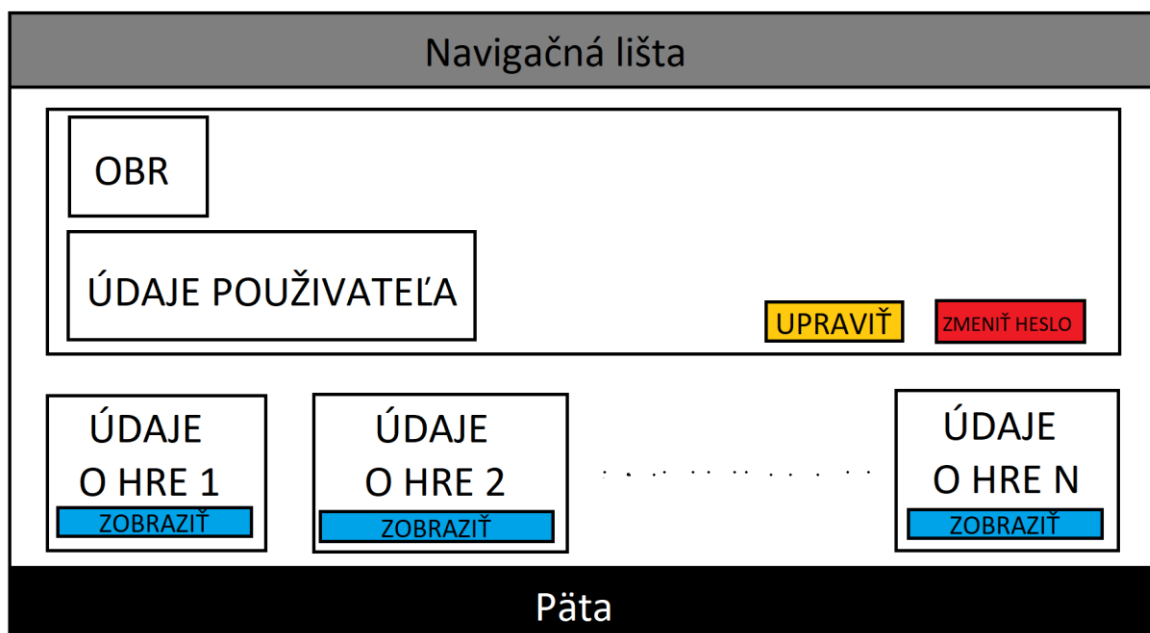
Päta

Obr. 34 - Dizajn podstránky zoznamu používateľov

3.4.10. Návrh a dizajn podstránky používateľa

Podstránka používateľa slúži ako informačný kanál pre samotného používateľa ako aj administrátora. Umožňuje zobrazenie registračných údajov ako prihlasovacie meno, meno a priezvisko, e-mailová adresa a rok narodenia. Používateľ si môže zmeniť svoje údaje a taktiež aj zmeniť heslo. Administrátor má tieto funkcie zakázané, aby sa nenarušila ochrana súkromných údajov. Ako ďalšia funkcia je udržiavanie záznamov o odohratých hrách, kde je možné analyzovať odohratú hru a vyvodiť z nej individuálne výsledky.

Dizajn môžeme vidieť na Obr. 35. Podstránka sa skladá z dvoch sekcií, kde v prvej sekcií sa nachádza obrázok (profilová fotka) používateľa a pod ňou sú zobrazené údaje o používateli. Na pravej strane sú tlačidlá na úpravu údajov používateľa. Údaje používateľa je možné upraviť pomocou formulára podobného pri registrovaní nového používateľa s rozdielom, že nie je možné zmeniť heslo a používateľské meno. Druhé tlačidlo umožňuje zmeniť heslo pomocou formulára. Pod sekciou s používateľskými údajmi sa nachádza sekcia so štatistickými údajmi odohraných hier. Každý údaj obsahuje názov hry, dátum odohrania a získaný počet bodov. Tlačidlo „ZOBRAZIŤ“ slúži na zobrazenie detailnejšieho zoznamu, kde sú zobrazené konkrétne štatistiky.

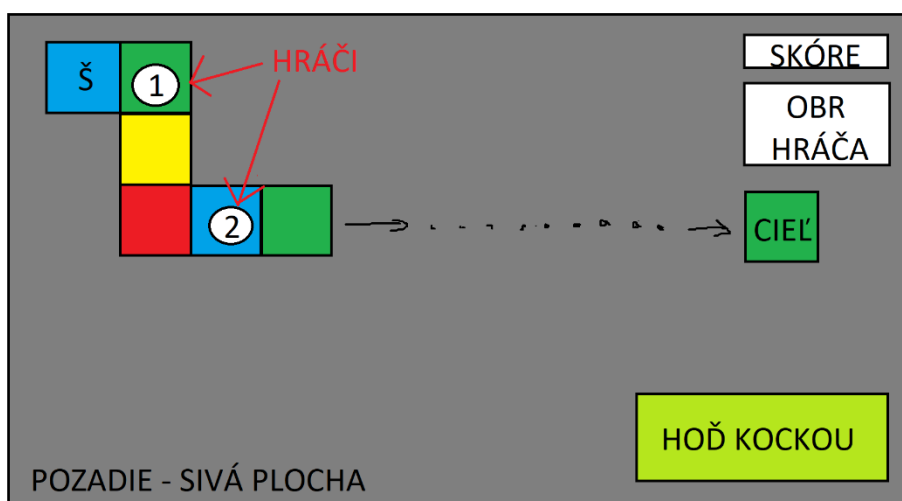


Obr. 35 - Dizajn podstránky používateľa

3.4.11. Návrh a dizajn podstránky hry

Najdôležitejšia a najzložitejšia podstránka platformy. Hlavným účelom podstránky je vygenerovať herné pole, po ktorom sa môže hráč pohybovať a plniť úlohy. Dostupné sú dva režimy podľa počtu hráčov. Hra pre jedného hráča umožňuje zbierať body počas postupu po hracom poli. Hráč postupuje po hracom poli pomocou hodu kockou, kde nadobúda hodnoty 1 až 6. Pri správnej odpovedi získava body, pri nesprávnej odpovedi nezískava body. Na konci hry sa uložia výsledky do databázy, ktoré sa následne dajú zobrazíť ako boli vysvetlené v odseku 3.4.10. Režim pre dvoch hráčov umožňuje hrať dvom hráčom naraz. Zmenou oproti režimu pre jedného hráča je v pohybe po hracom poli a získavanie bodov. Cieľom v režime pre dvoch hráčov je poraziť druhého hráča v pretekoch do cieľa. Pohyb po hracom poli sa vykonáva podľa správnej odpovede na otázku. Každý druh otázky má iný počet skokov na poli, nesprávna odpoveď nedáva možnosť hráčovi ďalší pohyb po pomaly pokiaľ neodpovie správne na otázku. Po ukončení hry sa uložia výsledky prvého hráča do databázy. Otázky sú vygenerované zo skupiny, ktorú administrátor vopred pripravil. Pevne stanovené sú iba okruhy otázok.

Dizajn hry je veľmi variabilný a dynamický. Záleží aký tvar herného poľa administrátor vytvoril a vložil do konkrétnej hry. Základ tvoria herné pole zložené zo štvorcov, ktoré sú farebne rozdielne. Každá farba reprezentuje iný typ úlohy, ktorý sa následne po hode kockou zobrazí pomocou vyskakovacieho okna. Dizajn hráčov opäť závisí od voľby administrátora. Skóre hráča je zobrazené v pravom hornom rohu obrazovky. Ak je zvolený režim pre dvoch hráčov namiesto hodnoty skóre sa vypíše, ktorý hráč je na ťahu a obrázok jeho hernej postavy, aby tomu porozumeli deti. Následne v pravom dolnom rohu je umiestnené tlačidlo pre hod kockou, ktoré je viditeľné iba pre režime jedného hráča. Na záver pozadie závisí od nastavenia administrátora. Na Obr. 36 môžeme vidieť princíp dizajnu.



Obr. 36 - Dizajn hry

3.5. Vytvorenie databázy platformy

Databáza tvorí najdôležitejšiu časť platformy, uchovávanie údajov. Skladá sa zo 14 tabuliek, ktoré vieme rozdeliť do troch veľkých sekcií, zobrazené na Obr. 38:

- nosné tabuľky,
- herné tabuľky,
- používateľské tabuľky.

3.5.1. Nosné tabuľky

Sekciu nosných tabuliek (červená sekcia s označením 1.) tvoria tabuľky datasets a games. Pre vytvorenie funkčnej databázy bolo potrebné, aby boli vytvorené ako prvé z dôvodu nadväznosti na ďalšie tabuľky. Tabuľka datasets obsahuje názov datasetu a identifikačné číslo, ktorý slúži v iných tabuľkách ako cudzí kľúč. Tabuľka games obsahuje základné informácie o hre (názov, priečinok hry), spomínaný cudzí kľúč z tabuľky datasets a verejnosť hry. Pomocou nosných tabuliek je možné vytvoriť prepojenie medzi hernými a používateľskými tabuľkami.

3.5.2. Herné tabuľky

Herné tabuľky (zelená sekcia s označením 2.) sú tvorené siedmimi tabuľkami, rozdelené na prepojovacie a dátové tabuľky. Tabuľka picture_category obsahuje názov kategórie a identifikačné číslo, ktoré sa používa ako cudzí kľúč pre prepojenie tabuľky pictures. Tabuľka pictures v sebe zahŕňa názov obrázka, jeho kategóriu a cesta ku súboru. Tabuľka riddles obsahuje stĺpce pre znenia hádaniek a odpovede na nich. Tabuľka simple_words ukladá len jednoduché slová. Následne sa využívajú prepojovacie tabuľky: dataset_pictures, dataset_riddles a dataset_simple_words. Prepojovacie tabuľky slúžia na prepojenie s tabuľkou datasets, medzi ktorými potrebujeme vytvoriť vzťah M:N. Obsahujú hodnoty cudzích kľúčov tabuliek a tak vytvárajú väzbu medzi tabuľkami.

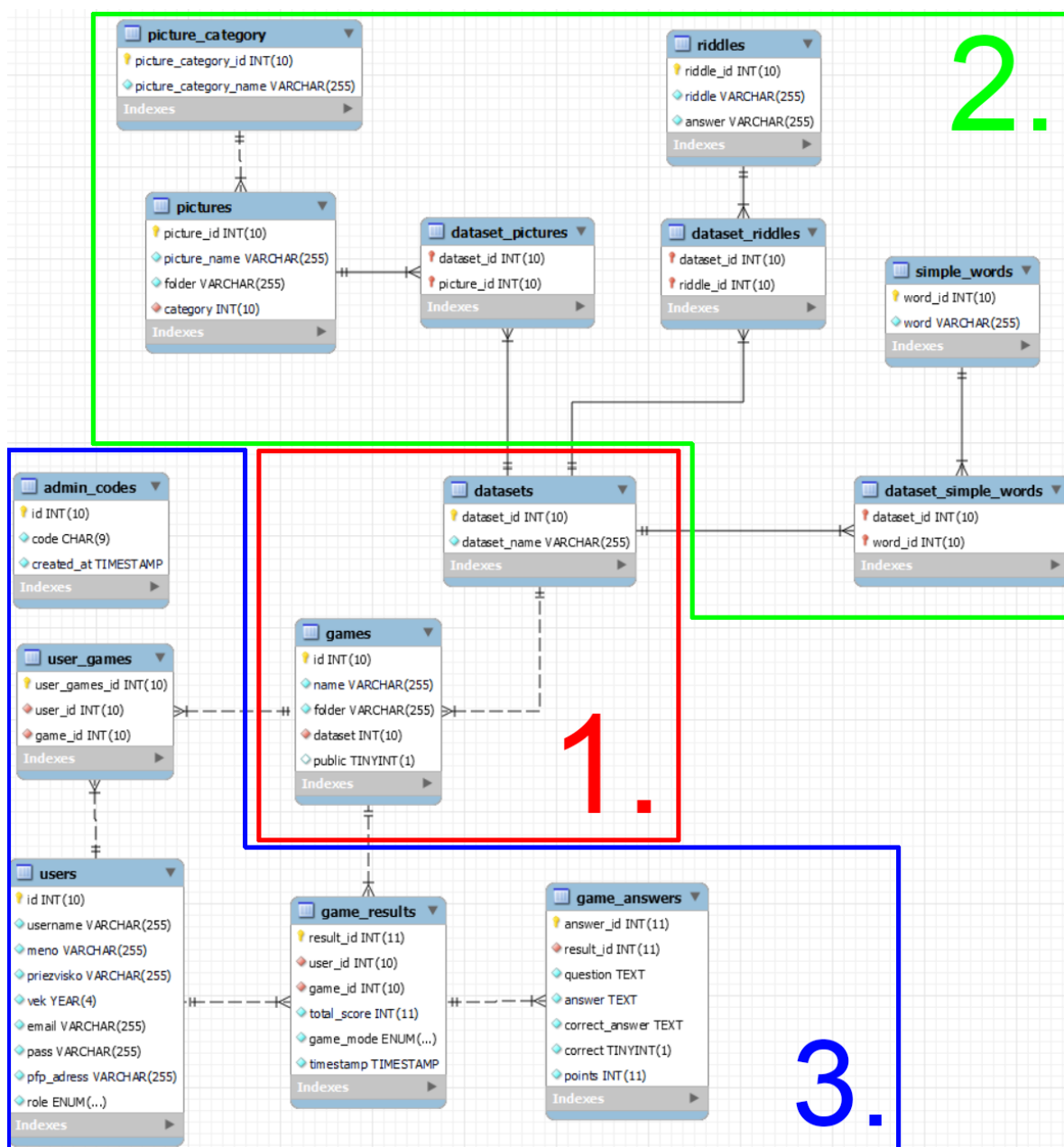
3.5.3. Používateľské tabuľky

Používateľské tabuľky (modrá sekcia s označením 3.) obsahujú päť tabuliek. Tabuľky majú za úlohu spravovať používateľské údaje a výsledky. Hlavnou tabuľkou v sekcii je tabuľka users, ktorá je tvorená informáciami o zaregistrovaných používateľoch (prihlasovacie meno, meno, priezvisko, e-mailová adresa, rok narodenia). Heslo pre bezpečnostné dôvody nie je uložené v tabuľke, ale iba jeho odtlačok (hash), ktorý bol vygenerovaný pomocou algoritmu bcrypt. Na záver tabuľky sa ukladá adresa profilovej fotografie používateľa a jeho rola v platforme (user, admin, root). Podľa role sa určujú administrátorské práva pre používateľa. Tabuľka admin_codes zaznamenáva jednorazové 9-miestne čísla, ktoré umožňujú vytvoriť nový administrátorský účet. Na Obr. 37 môžeme vidieť vytváranie nového kľúča pomocou SQL príkazu INSERT.

```
INSERT INTO admin_codes (code)
VALUES
(LPAD(FLOOR(RAND() * 1000000000), 9, '0'));
```

Obr. 37 - Vytváranie nového administrátorského kľúča

Ďalšou tabuľkou je `user_games`, ktorá plní funkciu prepojujúcej tabuľky s tabuľkou `games`. Ako posledné tabuľky v sekcii sú `game_results` a `game_answers`. `Game_results` plní funkciu uchovávanía výsledkov odohratých hier. Jej identifikačné číslo sa používa v tabuľke `game_answers`, ktorá obsahuje konkrétne úlohy v rámci odohratej hry a jej výsledky.



Obr. 38 - Náhľad databázy platformy

```
CREATE TABLE users (
    id INT UNSIGNED PRIMARY KEY AUTO_INCREMENT,
    username VARCHAR(255) UNIQUE NOT NULL,
    meno VARCHAR(255) NOT NULL,
    priezvisko VARCHAR(255) NOT NULL,
    vek YEAR NOT NULL,
    email VARCHAR(255) UNIQUE NOT NULL,
    pass VARCHAR(255) NOT NULL,
    pfp_adress VARCHAR(255) NOT NULL DEFAULT 'default-avatar.png',
    role ENUM('user', 'admin', 'root') NOT NULL DEFAULT 'user'
);
```

Obr. 39 - Ukážka príkazu CREATE TABLE

3.6. Hierarchia súborov

Pred začatím vysvetľovania implementácií musíme zdefinovať pojmy, na ktoré sa budeme odkazovať z dôvodu jasnejšieho pochopenia. V administrátorskej časti budeme pracovať so súbormi, ktoré budeme označovať súbor 1, 2 a 3. Pre jasnejšie pochopenie, v akom súbore sme pracovali môžeme vidieť v Tab. 1. Súbor 1 bude mať za úlohu zobrazenie podstránky, súbor 2 funkcionálnu časť a prepojenie so stranou servera pomocou AJAX a súbor 3 obsahuje databázové dopyty a príkazy, ktoré pracujú na serveri.

Podstránka	Názov súboru 1	Názov súboru 2	Názov súboru 3
Správa kategórií obrázkov	manage_categories.php	admin_categories.js	manage_categories_ajax.php
Správa obrázkov	manage_images.php	admin_images.js	manage_images_ajax.php
Správa úloh	manage_tasks.php	admin_tasks.js	manage_tasks_ajax.php
Správa datasetov	manage_datasets.php	admin_datasets.js	manage_datasets_ajax.php
Správa hier	manage_games.php	admin_games.js	manage_games_ajax.php
Správa používateľov	manage_users.php	admin_users.js	manage_users_ajax.php
Profil používateľa	user_profile.php	user_profile.js	user_profile_ajax.php

Tab. 1 - Súbory v administrátorskej časti

3.7. Implementácia registrácie a prihlasovania

Po vytvorení databázy je potrebné vytvoriť funkcionality registrácie a prihlásenia. Funkcionalita zabezpečí prístup ku funkciám, ktoré si vyžadujú prihlásených používateľov alebo používateľov s vyšším stupňom právomocí.

3.7.1. Vytvorenie súboru login.js

Pre zjednodušenie a zníženie redundancie bol vytvorený súbor login.js, ktorý obsahuje vyskakovacie okná umožňujúce registráciu, prihlásenie a taktiež prepojenie s backend časťou platformy. Súbor bude jednoducho pripojený ku každej PHP podstránke, kde sa bude vyžadovať. Súbor reaguje na tlačidlá „Prihlásiť“ a „Registovať“ umiestnené v navigačnej lište. Vykreslenie formulárov bude vykonávané cez jQuery príkaz `$("body").append(``), do ktorého je možné vložiť HTML elementy.

3.7.2. Vytvorenie formulára pre registráciu

Na začiatok je potrebné vytvoriť formulár pre registráciu. Podľa návrhu z odseku 3.4.1 bol vytvorený formulár pomocou HTML tagov `<form>`, do ktorého boli pridané vstupné elementy `<input>`. Vstupné elementy majú nad sebou textové označenie, ktoré popisuje význam elementu. Ku každému elementu bolo potrebné nastaviť pravidlá pre vstupné hodnoty, aby sa predišlo nekonzistenciám dát:

- **Prihlasovacie meno:** minimálna dĺžka 5 znakov,
- **Meno:** minimálna dĺžka 3 znakov a povolené písmena v slovenskej abecede,
- **Priezvisko:** minimálna dĺžka 3 znakov a povolené písmena v slovenskej abecede,
- **Rok narodenia:** interval <1900, aktuálny rok>,
- **E-mailová adresa:** riešená v JS (vysvetlené v odseku 3.7.4),
- **Heslo:** minimálna dĺžka 8 znakov, aspoň 1 malé a veľké písmeno, aspoň 1 číslo.

```
<!-- MENO -->
<div class="col-md-6 mb-3">
  <label for="meno" class="form-label">Meno</label>
  <div class="input-group">
    <span class="input-group-text"><i class="fa fa-id-badge"></i></span>
    <input type="text" class="form-control" id="meno" name="meno" required pattern="[a-zA-ZäöčďéěíĺľňóôřšťúůýžÁĀČĎĚĚÍĹĽŇŌŌŘŠŤÚŮÝŽ ]+" minlength="3"
  </div>
</div>
```

Obr. 40 - Ukážka vytvorenia vstupného elementu pre meno

3.7.3. Vytvorenie formulára pre prihlásenie

Proces bol podobný ako v odseku 3.7.2, kde bol vytvorený formulár pre prihlásenie, ktorý obsahuje vstupné elementy pre prihlasovacie meno alebo e-mailovú adresu. Pod týmito elementami bola pridaná možnosť „Ostať prihlásený“, ktorú vysvetlíme v odseku 3.7.6.

3.7.4. Prepojenie formulárov s backendom

Po vyplnení formulárov bolo potrebné zaistiť, aby sa údaje preposlali na server a vykonali operácie určené podľa typu funkcie. Ak používateľ klikne na tlačidlo pri registrácii, súbor zareaguje na tlačidlo so zadaným identifikátorom „#register-form“. Následne sa vykoná posledná kontrola, ktorá nemohla byť vykonaná v HTML kóde s dôvodu chyby preddefinovaného reťazca na kontrolu (regex).

Chyba je však na strane prehliadača, tak sme sa to rozhodli vykonať v JavaScript súbore. Ak kontrola prebehne úspešne, vytvorí sa nová premenná typu FormData, ktorá umožní poslať dáta z formulára na spracovanie. Posielanie dát je umožnené pomocou AJAX a REST API dopytu POST, ktorý umožní poslať dáta na server. AJAX plní úlohu asynchrónneho spracovania, čo umožňuje spracovanie dát bez potreby obnovovania podstránky. AJAX pri registrácii zavolá súbor signup.php, ktorý zabezpečí uloženie údajov do databázy a následne odpovie späť JS súboru, či bola transakcia úspešná alebo nebola. Pri prihlasovaní je proces rovnaký s rozdielom reagovania súboru na tlačidlo s identifikátorom „#login-form“ a AJAX volaním súboru login.php.

```
$.ajax({  
    url: "php/signup.php",  
    type: "POST",  
    data: formData,  
    processData: false, // Zabrániť jQuery  
    contentType: false, // Necháme prehliadač  
    success: function (response) :void {  
        let data = JSON.parse(response);  
    }  
});
```

Obr. 41 - Príklad použitia AJAX

3.7.5. Spracovanie registračných dát

Súbor signup.php zabezpečuje zápis užívateľa do databázy. Na začiatok bol vytvorený súbor connection.php, ktorý vytvorí spojenie s databázou, ktorá bola vytvorená v odseku 3.5. Súbor nám posluží aj v ďalších súboroch pre spracovanie údajov.

```
<?php  
  
$servername = "localhost";  
$username = "root";  
$password = "";  
$dbname = "logopedy_game";  
  
$conn = mysqli_connect($servername, $username, $password, $dbname);  
if (!$conn) {  
    die("Pripojenie k databáze zlyhalo: " . mysqli_connect_error());  
}  
  
return $conn;
```

Obr. 42 - Vytvorenie spojenia s databázou

Po vytvorení spojenia boli načítané údaje, ktoré nám boli odoslané pomocou metódy POST. Ako ďalší krok je potrebné overiť, či používateľské meno a e-mailová adresa nie je použitá. To bolo realizované cez ďalší súbor s názvom find_existing.php, v ktorom sa pomocou SQL dopytu SELECT COUNT(*) vyhladá počet záznamov v tabuľke users pre danú e-mailovú adresu a prihlasovacie meno. Ak sa nájde záznam, tak registrácia nepokračuje a vypíše sa používateľovi, že registrácia je neúspešná.

```
if ($_POST["new_user"]) {  
    $stmt = $conn->prepare("SELECT COUNT(*) FROM users WHERE email = ?");  
    $stmt->bind_param("s", $email);
```

Obr. 43 - SQL príkaz na kontrolu e-mailovej adresy



Chyba!

Toto prihlasovacie meno existuje.

OK

Obr. 44 - Neúspešná registrácia s dôvodu existujúceho prihlasovacieho mena

Po overení bolo potrebné ošetriť ukladanie profilovej fotografie na server, ak bola nastavená. Ak užívateľ zvolil novú profilovú fotografiu, nastavili sme, aby sa ukladala do priečinka „assets/user_profile_pictures/“ pod používateľským meno používateľa. Ak užívateľ nenahral vlastnú fotografiu, použije sa vopred definovaná fotografia. Po vložení fotografie sa uloží do premennej názov súboru a prípona, ktorá bude použitá pri vytváraní záznamu do databázy.

Ďalším krokom bolo overenie administrátorského kľúča. Ak užívateľ požiada o vytvorenie administrátorského účtu, overí sa platnosť kľúča pomocou SQL dopytu SELECT z tabuľky admin_codes. Ak sa takýto kľúč nachádza v tabuľke, je verifikovaný a daný kľúč je vymazaný z tabuľky pomocou SQL príkazu DELETE, čím sa eliminuje jeho duplicitné využitie, poprípade zneužitie kľúča.

```

if ($result->num_rows > 0) {
    // Ak je kód platný, nastavíme rolu 'admin' a kód vymažeme
    $role = "admin";
    $stmt = $conn->prepare("DELETE FROM admin_codes WHERE code = ?");
    $stmt->bind_param("s", $adminCode);
    $stmt->execute();
} else {
    // Ak je kód nesprávny, vrátime chybu
    echo json_encode(["success" => false, "message" => "Nesprávny admin kód!"]);
    exit;
}

```

Obr. 45 - Overenie kľúča

Po poslednom overení bol vytvorený SQL príkaz INSERT, ktorý umožní vložiť hodnoty do tabuľky users a tým zaregistrovať používateľa. Vďaka všetkým overovaniám sme docielili elimináciu duplicity, nekonzistentných a chybných údajov. Na záver sa nastaví hodnoty do \$_SESSION, ktoré slúžia na neskoršie spracovanie a taktiež overenie, či je už používateľ prihlásený. Ak všetko prebehlo úspešne vráti odpoveď do JS súboru, ktorý vyhodnotí registráciu za úspešnú a obnoví stránku.

```

if (isset($_FILES["profile_image"]) && $_FILES["profile_image"]["error"] === UPLOAD_ERR_OK){
    $stmt = $conn->prepare("INSERT INTO users (username, meno, priezvisko, vek, email, pass, pfp_adress, role) VALUES (?, ?, ?, ?, ?, ?, ?, ?)");
    $stmt->bind_param("sssssss", $username, $meno, $priezvisko, $vek, $email, $hashedPassword, $newFileName, $role);
}

```

Obr. 46 - Nahratie údajov do tabuľky users

3.7.6. Prihlásenie používateľa

V súbore login.php sme vytvorili jednoduchý dopyt, ktorý skontroluje vstupné údaje, ktoré boli vložené do formulára pomocou SQL dopytu SELECT z tabuľky users. Ak sa nájde používateľ, vieme s jednoznačnosťou povedať, že ide o konkrétného používateľa. Následne sa vykoná verifikácia hesla, ktorá po úspešnom overení nastaví hodnoty do \$_SESSION a pošle správu do JS súboru. Následne podľa hodnoty \$_SESSION['stay_logged_in'] sa overí, či je hodnota true, teda či používateľ zvolil možnosť ostať prihlásený. Ak áno, používateľ je prihlásený na dobu 1 mesiaca (1 hodiny ak nie).

3.8. Implementácia kategórií obrázkov

Po vytvorení registrácie a prihlasovania boli postupne vytvárané základne podstránky z administrátorského pohľadu. Prvou podstránkou bola správa kategórií obrázkov. Podľa odseku 3.4.8 sme vytvorili súbor 1, kde sme implementovali dizajn formulára na pridávanie nových kategórií.

Ako ďalší krok bol pripravený HTML kód, ktorý bude dynamicky reagovať na vytvorené kategórie v databáze a bude ich vypisovať.

Pomocou SQL dopytu SELECT na Obr. 47, sa načítajú všetky kategórie a počet zaradených obrázkov v danej kategórii.

```
// Načítanie kategórií
$query = "SELECT c.picture_category_id, c.picture_category_name,
           COUNT(p.picture_id) AS image_count
FROM picture_category c
LEFT JOIN pictures p ON c.picture_category_id = p.category
GROUP BY c.picture_category_id";
```

Obr. 47 - SQL dopyt na zobrazenie kategórií

Do výpisu boli implementované aj tlačidlá na úpravu a vymazanie, ktoré majú pridelené číselné identifikátory kategórií, aby bolo jednoznačne určené, na ktorej kategórii sa majú vykonať zmeny. Všetky reakcie sú umiestnené v súbore 2. Súbor obsahuje funkcie: pridanie, úpravu a jej potvrdenie a odstránenie kategórie. Všetky tieto funkcie spolupracujú so súborom 3 pomocou AJAX-u. Po vykonaní operácií na strane servera sa prečíta odpoveď a vygeneruje vyskakovacie okno, ktoré upozorní používateľa o úspechu resp. neúspechu.

Pri pridávaní novej kategórie bolo potrebné implementovať opätovné vykreslenie novo pridanej kategórie z dôvodu, kde pôvodný kód umiestnený v súbore 1 reaguje iba pri načítavaní stránky. Týmto spôsobom je eliminovaná potreba obnovovania stránky. Podobným spôsobom boli riešené aj funkcie pre odstránenie a úpravu. Pri odstraňovaní bola využitá jQuery funkcia remove(), ktorá predtým nájde požadovaný element s identifikátorom kategórie a odstráni ho využitím 0,5 sekúnd dlhej animácie fadeOut(). Úprava bola vyriešená jednoduchým prepísaním textu za novú hodnotu.

```
$("#category-" + categoryId).fadeOut(500, function () :void {
    $(this).remove();
});
```

Obr. 48 - Plynulé odstránenie kategórie

```
categoryItem.find(".category-name").text(newName);
```

Obr. 49 - Úprava názvu kategórie

```

let response = JSON.parse(data);
let newCategory :string = `<div class="col-md-4 col-sm-6 mb-3" id="category-${response.id}">
<div class="card shadow-sm">
  <div class="card-body">
    <div class="d-flex justify-content-between align-items-center">
      <span class="category-name">${response.name}</span>
      <span class="badge bg-primary">0 obrázkov</span>
      <input type="text" class="form-control edit-category-name" value="${response.name}" style="...">
    </div>
    <div class="mt-2 d-flex justify-content-end">
      <button class="edit-btn btn btn-warning btn-sm me-2" data-id="${response.id}">
        <i class="fa fa-pencil"></i> Upraviť
      </button>
      <button class="save-btn btn btn-success btn-sm me-2" data-id="${response.id}" style="...">
        <i class="fa fa-check"></i> Uložiť
      </button>
      <button class="delete-btn btn btn-danger btn-sm" data-id="${response.id}">
        <i class="fa fa-trash"></i> Vymazať
      </button>
    </div>
  </div>
</div>
</div>`;
$("#category-list").append(newCategory);

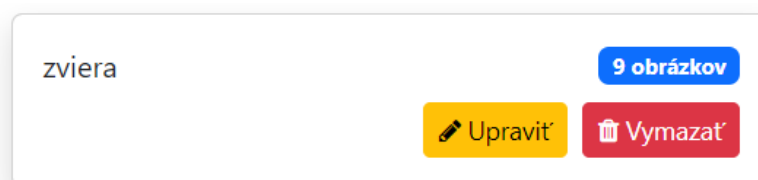
```

Obr. 50 - Zobrazenie novej kategórie po odpovedi na AJAX

Do súboru 3 boli vytvorené tri funkcie, ktoré pracujú s databázou. Každá z funkcií reaguje na iný typ akcie. Pri AJAX dopyte sa posiela v hodnote \$action určité slovo. Podľa týchto slov reaguje súbor nasledovne:

- **\$action == „add“** -> pridanie záznamu do databázy,
- **\$action == „edit“** -> úpravy záznamu v databáze,
- **\$action == „delete“** -> odstránenie záznamu z databázy.

Následne boli vytvorené tri rôzne SQL príkazy, ktoré sa majú vykonať pre tabuľku picture_categories. Pre pridanie je to SQL príkaz INSERT, pre úpravu záznamu UPDATE a pre odstránenie príkaz DELETE. Úprava a odstránenie využívajú identifikátor kategórie ako podmienku pre korektné vykonanie príkazu. Výsledok môžeme vidieť na Obr. 51, ktorý sa zobrazí, ak záznam existuje v tabuľke picture_categories.



Obr. 51 - Výpis kategórie obrázkov

3.9. Implementácia obrázkov

Podobne ako v prechádzajúcom bode, bol na začiatok vytvorený nový súbor 1. V súbore boli vytvorený formulár pre pridanie nových obrázkov pozostávajúci z názvu obrázka, samotného súboru vo formáte JPG alebo PNG a kategórie obrázka. Kategórie obrázkov sa načítavajú do elementu <select> pod možnosti <option>. Medzi tieto možnosti bol implementovaný SQL dopyt `SELECT * FROM picture_category`, ktorý vyhľadá všetky existujúce kategórie v tabuľke. Následne bol výstup vložený do cyklu `while`, ktorý sa vykonáva pokiaľ existuje záznam.

```
<?php
$categories = $conn->query("SELECT * FROM picture_category");
while ($cat = $categories->fetch_assoc()) {
    echo "<option value='{ $cat['picture_category_id'] }'>{ $cat['picture_category_name'] }</option>";
}
}
```

Obr. 52 - Výpis kategórií obrázkov

Ako ďalší krok bolo vytvorenie veľkého elementu <div>, do ktorého sa budú načítavať existujúce obrázky. Výpis je realizovaný vďaka SQL dopytu `SELECT` zobrazenom na Obr. 53. Dopyt načíta všetky obrázky spolu s ich parametrami (identifikačné číslo, meno, priečinok umiestnenia a kategóriu). Následne sa podobným spôsobom vykoná vykreslenie pomocou cyklu `while`. Ku tomuto výpisu bol pridaný formulár na úpravu, do ktorého sa preddefinujú aktuálne hodnoty.

```
// Načítanie všetkých obrázkov
$query = "SELECT p.picture_id, p.picture_name, p.folder, c.picture_category_name
FROM pictures p
LEFT JOIN picture_category c ON p.category = c.picture_category_id";
```

Obr. 53 - Výpis obrázkov

Ako ďalší súbor bol vytvorený súbor 2, ktorý rieši funkcie. Opäť súbor riešil funkcie pridania, úpravy a odstraňovania obrázka. Postupovalo sa rovnako ako v predošlom bode z rozdielom ako sa vykonávajú zmeny po akcii. Hlavnou zmenou, ktorú bolo potrebné vyriešiť, bola úprava obrázka. Keďže v prípade zmeny obrázka sa môže meniť aj samotný obrázok bolo potrebné vytvoriť plynulý efekt zmeny. To bolo docielené, že v súbore 3, o ktorom si povieme detailnejšie neskôr vytvárame nový element pre výpis obrázka, ktorý následne vráti do JS súboru ako odpoveď. Táto odpoveď sa načíta a nahradí sa novou. Efekt je docielený, že pomocou funkcie `animate()` je po dobu jednej sekundy pôvodnému bloku nastavená hodnota opacity na 0 (stane sa neviditeľným) a následne sa vykonaná zmena pomocou nahradenia starého bloku za nový. Poslednú vec, ktorú bolo potrebné ošetriť, bola zmena zdrojovej cesty pre obrázok. Pri vytváraní bol zistený istý problém s prehliadačom, kde ak bol zmenený súbor, ale jeho názov ostal nezmenený, tak prehliadač stále

načítaval pôvodný obrázok, ktorý mal uložený v cache pamäti. Zmena sa však eliminovala obnovením stránky. Pre zachovanie funkcie stránky bez obnovovania bol problém vyriešený pridaním časovej značky (timestamp) ku zdrojovej ceste. Časová značka nenaruší načítanie obrázka a vygeneruje sa iba pri úprave, čím bol docieľený požadovaný efekt. Následne je bloku nastavená hodnota opacity na 1 (obrázok je zase viditeľný) a opäť pomocou funkcie animate() trvajúcej 1 sekundu je vytvorený efekt postupného zviditeľnenia bloku.

```
// Pridaj časovú značku k URL obrázka
var src = newImageElement.attr("src");
src = src.replace(/\?t=\d+$/, "");
newImageElement.attr("src", src + "?t=" + new Date().getTime());
```

Obr. 54 - Vyriešenie problému s načítaním obrázka

Ako tretí bol vytvorený súbor 3. Podobne ako v predošlom kroku reaguje na tri akcie, ktoré sú posielané cez AJAX spolu s dátami pre spracovanie:

- **\$action == „add“** -> pridanie záznamu do databázy,
- **\$action == „edit“** -> úprava záznamu v databáze,
- **\$action == „delete“** -> odstránenie záznamu z databázy.

V tomto prípade však pribudol nový problém, ktorý bolo potrebné vyriešiť a tým je uloženie obrázka. Bolo potrebné vyriešiť ako uložiť obrázok bez nutnosti ukladania do databázy. Riešením vzniklo uloženie obrázka na samotný server do adresára „assets/sada“ a uloženie jeho cesty do databázy, čím bola značne zredukovaná veľkosť a rýchlosť databázy. Na začiatku načítame názov, ktorý používateľ zadefinuje a pod ktorým sa bude tento súbor volať na serveri, aby sme predišli nekonzistentným názvom. Po vytvorení potrebných názvov sa súbor nahrá na server pomocou príkazu move_uploaded_file(), do ktorého vložíme samotný obrázok a nový názov. Ak uloženie prebehlo úspešne, uložíme meno, priečinok a kategóriu obrázka do tabuľky pictures. Následne vytvoríme nový blok obrázka pre výpis a pošleme ho ako odpoveď pre AJAX, ktorý ho zobrazí.

```
if (move_uploaded_file($_FILES["file"]["tmp_name"], "../" . $file_path)) {
    $stmt = $conn->prepare("INSERT INTO pictures (picture_name, folder, category) VALUES (?, ?, ?)");
    $stmt->bind_param("ssi", $picture_name, $file_path, $category);
    $stmt->execute();
}
```

Obr. 55 - Uloženie nového obrázka

Pri úprave obrázka bolo potrebné ošetriť stav, kde ak používateľ vloží nový obrázok, je potrebné, aby sa pôvodný obrázok odstránil. Tento stav bol ošetrený načítaním pôvodných hodnôt obrázka pomocou SQL dopytu SELECT folder FROM pictures WHERE picture_id = ?, ktorý vyhľadá obrázok.

Po načítaní bola využitá funkcia `unlink()`, ktorá odstráni pôvodný súbor, ktorý bol vyhľadáný a nahrá sa nový súbor pomocou funkcie `move_uploaded_file()`. Následne sa vykoná SQL príkaz `UPDATE`, ktorý aktualizuje pôvodné hodnoty obrázka. Ako posledný krok sa opäť vytvorí aktualizovaný blok, ktorý sa zobrazí pomocou JS súboru ako odpoveď pre AJAX.

```
// Načítanie aktuálnych údajov o obrázku
$stmt = $conn->prepare("SELECT folder FROM pictures WHERE picture_id = ?");
$stmt->bind_param("i", $picture_id);
$stmt->execute();
$result = $stmt->get_result();
$row = $result->fetch_assoc();
$old_file_path = "../" . $row['folder'];

// Spracovanie nového obrázka, ak bol nahraný
if (!empty($_FILES["new_image"]["name"])) {
    $picture_name_save = preg_replace( pattern: '/[^\a-zA-Z0-9_-]/', replacement: '', $picture_name);
    $new_file_name = $picture_name_save . '.' . pathinfo($_FILES["new_image"]["name"], flags: PATHINFO_EXTENSION);
    $new_file_path = "assets/sada/" . $new_file_name;

    if (unlink($old_file_path)){
        move_uploaded_file($_FILES["new_image"]["tmp_name"], "../" . $new_file_path);
        $stmt = $conn->prepare("UPDATE pictures SET picture_name = ?, folder = ?, category = ? WHERE picture_id = ?");
        $stmt->bind_param("ssii", $picture_name, $new_file_path, $category, $picture_id);
    } else {
        echo json_encode(["success" => false, "message" => "Nepodarilo sa nahrať nový obrázok"]);
        exit;
    }
} else {
    $stmt = $conn->prepare("UPDATE pictures SET picture_name = ?, category = ? WHERE picture_id = ?");
    $stmt->bind_param("sii", $picture_name, $category, $picture_id);
}
```

Obr. 56 - Úprava obrázku

Odstránenie obrázka bola najjednoduchšia funkcia z trojice, pretože nebolo potrebné riešiť žiadne nové podmienky a ošetrenia. Odstránenie prebehne načítaním cesty ku obrázku, ktorý odstránime pomocou funkcie `unlink()`, ak súbor existuje a následne vymažeme záznamy z databázy pomocou SQL príkazu `DELETE`.

```
} elseif ($_POST["action"] == "delete") {
    $picture_id = $_POST["picture_id"];
    $stmt = $conn->prepare("SELECT folder FROM pictures WHERE picture_id = ?");
    $stmt->bind_param("i", $picture_id);
    $stmt->execute();
    $result = $stmt->get_result();
    $row = $result->fetch_assoc();

    if ($row) {
        $file_path = "../" . $row["folder"];
        if (file_exists($file_path)) {
            unlink($file_path); // Vymazanie súboru
        }
    }

    $stmt = $conn->prepare("DELETE FROM pictures WHERE picture_id = ?");
    $stmt->bind_param("i", $picture_id);
    $stmt->execute();
}
```

Obr. 57 - Odstránenie obrázka



The image shows three example cards from a database. Each card contains an illustration, a name, a category, and two buttons: 'Upraviť' (Edit) and 'Vymazať' (Delete).

- Card 1:** Illustration of a bunch of purple grapes. Name: **hrozno**. Category: **Kategória: ovocie**. Buttons: 'hrozno', 'ovocie', 'Vybrať súbor', 'N...', 'Upraviť', 'Vymazať'.
- Card 2:** Illustration of a red strawberry. Name: **jahoda**. Category: **Kategória: ovocie**. Buttons: 'jahoda', 'ovocie', 'Vybrať súbor', 'N...', 'Upraviť', 'Vymazať'.
- Card 3:** Illustration of a grey and white striped animal (likely a civet or mongoose). Name: **jazvec**. Category: **Kategória: zvierá**. Buttons: 'jazvec', 'zvierá', 'Vybrať súbor', 'N...', 'Upraviť', 'Vymazať'.

Obr. 58 - Ukážka výpisu obrázkov

3.10. Implementácia úloh

Ako posledný elementárny prvok pre fungovanie platformy bola implementovaná správa úloh, kde je možné spravovať hádanky a jednoduché slová. Bol vytvorený súbor 1, do ktorého boli vložené podobné formuláre a výpis existujúcich úloh z databázy. Na začiatok boli vytvorené SQL dopyty `SELECT * FROM riddles` a `SELECT * FROM simple_words`, ktoré získajú všetky hodnoty z tabuliek. Následne bol vytvorený formulár pre pridanie nových hádaniek. Štruktúra formulára je veľmi podobná z predošlých odsekov bez potreby ošetrovania vstupov. Pod formulárom bol zrealizovaný výpis hádaniek, ku ktorému boli pridané tlačidlá na úpravu a vymazanie. Podobným štýlom boli riešené aj jednoduché slová, kde sa iba zjednodušil formulár a dizajnovy zmenil výpis.

Súbor 2 obsahuje funkcie pre interakciu so serverom. Pracuje s funkciami pridania, úpravy a odstránenia hádaniek a jednoduchých slov. Hádanky a jednoduché slová majú svoje vlastné funkcie pretože vyžadujú iný prístup ku výpisu alebo prácu s databázou. Myšlienka funkcií ostáva totožná, ktorá pošle dáta pomocou AJAX-u na server a po vykonaní vypíše pridanú úlohu. Úprava a odstránenie fungujú na rovnakom princípe ako v predošlých odsekoch. Pri interakcií s tlačidlami sa objavujú vstupné elementy (pre úpravu) alebo vyskakovacie okno (pri odstraňovaní). Po vykonaní sa následne skontaktuje AJAX so serverom pre vykonanie úkonov, po ktorých sa vypíše zmena hodnôt.

Serverový súbor 3 obsahuje 6 funkcií, ktoré reagujú na tento typ akcie:

- `$action == „add_riddle“` -> pridanie hádanky do databázy
- `$action == „edit_riddle“` -> úprava hádanky v databáze
- `$action == „delete_riddle“` -> odstránenie hádanky z databázy
- `$action == „add_word“` -> pridanie slova do databázy
- `$action == „edit_word“` -> úprava slova v databáze
- `$action == „delete_word“` -> odstránenie slova z databázy

Toto rozdelenie bolo vykonané z dôvodu jednoznačného a prehľadnejšieho kódu, tak aby sa nedopatrením nepridali hodnoty do druhej tabuľky. Pridávanie bolo realizované pomocou SQL príkazu INSERT, úpravu pomocou UPDATE a vymazanie pomocou DELETE.

```
if ($action == "add_riddle") {  
    $riddle = $_POST["riddle"];  
    $answer = mb_strtolower($_POST["answer"]);  
    $stmt = $conn->prepare("INSERT INTO riddles (riddle, answer) VALUES (?, ?)");  
    $stmt->bind_param("ss", $riddle, $answer);  
    $stmt->execute();  
    echo json_encode(["id" => $stmt->insert_id, "riddle" => $riddle, "answer" => $answer]);  
}
```

Obr. 59 - Pridanie hádanky do databázy

Výsledkom implementácie je funkčná správa a výpis hádaniek a úloh, ktoré môžeme vidieť na Obr. 60 a Obr. 61.

Hádanka: Čo sa zamočí pri sušení?

Odpoveď: uterák

Obr. 60 - Výpis hádanky

krava  	rampa  
prst  	hra  

Obr. 61 - Výpis jednoduchých slov

3.11. Implementácia datasetov

Vďaka implementácií funkcií z odsekov 3.8, 3.9 a 3.10 ich teraz môžeme vložiť do jednej skupiny (datasetu). Správa datasetov je viac náročnejšia na funkcie ako predošlé odseky, preto rozdelíme túto implementáciu na viacero bodov a popíšeme ich ako po stránke dizajnovej, tak po stránke funkcionálnej a serverovej.

3.11.1. Výpis, tvorba a odstránenie datasetov

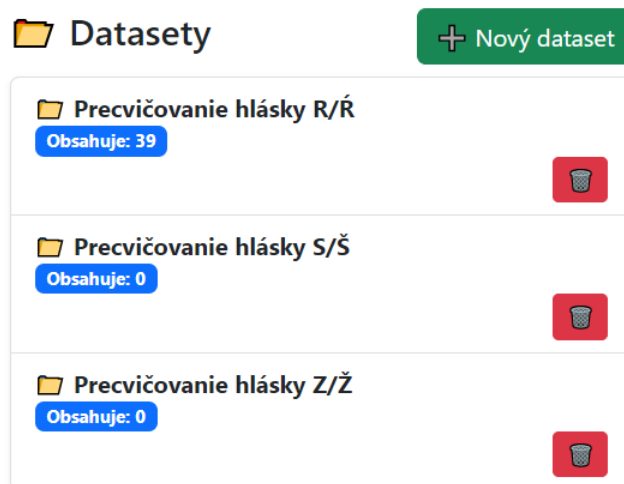
Výpis datasetov bol realizovaný na ľavej strane obrazovky, kde sa jednotlivé datasety zobrazujú pod sebou. V súbore 1 bol zadefinovaný element , ktorý reprezentuje neočíslovaný zoznam. Zobrazenie jednotlivých položiek je vykonávané pomocou funkcie loadDatasets() v súbore 2, ktorá kontaktuje súbor 3 pomocou AJAX dopytu na akciu get_datasets. V súbore 3 na túto akciu reaguje funkcia getDatasets(), do ktorej bol zadefinovaný špeciálny SQL dopyt na získanie identifikátora datasetu, názvu a počtu priradených úloh ku danému datasetu, vďaka súčtu hodnôt v prepojedných tabuľkách, kde sa nachádza identifikátor datasetu. Výstupom sú samotné výsledky, ktoré sú odoslané ako odpoveď pre AJAX.

```
$sql = "SELECT d.dataset_id, d.dataset_name,
        (SELECT COUNT(*) FROM dataset_simple_words WHERE dataset_id = d.dataset_id) +
        (SELECT COUNT(*) FROM dataset_riddles WHERE dataset_id = d.dataset_id) +
        (SELECT COUNT(*) FROM dataset_pictures WHERE dataset_id = d.dataset_id) AS item_count
FROM datasets d
ORDER BY dataset_name ASC";
```

Obr. 62 - SQL dopyt pre vyhľadanie datasetov

AJAX následne tieto výsledky spracuje a vloží do elementov, ktoré definujú obsah neočíslovaného zoznamu. Ako posledná vec pri výpise bola nastavená, aby sa pri prechode na konkrétny dataset kurzor myši zmenil na pointer.

Vytvorenie a zmazanie datasetu je veľmi jednoduchý proces, ktorý bol použitý v predošlých odsekoch. Vytvorenie je realizované pomocou vyskakovacieho okna, kde je možné zadať hodnotu. Hodnota následne je predaná pomocou AJAX-u na server pod akciou create_dataset. Pod akciou má súbor 3 zadefinovaný SQL príkaz CREATE. Odstránenie datasetu je realizované cez tlačidlo vymazávania, ktoré pomocou AJAX-u pošle na server akciu delete_dataset, ktorá realizuje SQL príkaz DELETE pre tabuľku datasets. Databáza je nastavená tak, aby sa pri odstránení datasetu odstránili aj všetky väzby na tento dataset a tými sú všetky záznamy v prepojedných tabuľkách s identifikátorom datasetu, ktoré sa odstránia automaticky.



Obr. 63 - Výpis datasetov

3.11.2. Výpis a odstránenie úloh v datasete

Pre každý dataset bolo potrebné umožniť výpis všetkých položiek. Riešenie bolo realizované pomocou kliknutia na dataset. Kliknutím na dataset sa načíta jeho identifikátor a nastaví sa nová trieda „active“, kde všetkým ostatným datasetom sa táto trieda odstráni. Triedou „active“ bolo zabezpečené jednoznačné získanie identifikátora aj pre ďalšie funkcie súvisiace s datasetom. Následne súbor 2 pošle AJAX dopyt s akciou `get_dataset_content`. Súbor 3 pod rovnomeným názvom funkcie vykoná tri SQL dopyty JOIN SELECT, ktoré nájdu všetky jednoduché slová, hádanky a obrázky, ktoré sú priradené v prepojediacich tabuľkách.

```
// Získanie jednoduchých slov v datasete
$wordsQuery = "SELECT sw.word, sw.word_id FROM dataset_simple_words dsw
                JOIN simple_words sw ON dsw.word_id = sw.word_id
                WHERE dsw.dataset_id = ?";
```

Obr. 64 - Ukážka SQL dopytu JOIN SELECT

Následne po získaní všetkých príslušných hodnôt, súbor vytvorí HTML elementy pomocou funkcie `echo()`, ktorá odpovedá priamo do AJAX-u a následne vykoná vykreslenie správ pomocou funkcie `$("#dataset-content").html(response)`. Implementoval sa taktiež filter na zobrazenie všetkých typov úloh alebo iba konkrétnych, kde podľa zvolenej hodnoty nájde zoznam, ktorý spĺňa podmienku a zobrazí pomocou funkcie `show()`. Ostatné zoznamy skryje pomocou funkcie `hide()`.

Po vypísaní sa zobrazia aj správčovské tlačidlá pre pridanie nových úloh do datasetu a ich odstránenie. Odstránenie prebieha podobným spôsobom ako v predošlých odsekoch pomocou SQL príkazu DELETE s rozdielom, že v tomto prípade sa odstraňujú údaje z prepojediacich tabuliek a ich korektné odstránenie vyžaduje identifikátor datasetu a konkrétnej úlohy.



Obr. 65 - Obsah datasetu

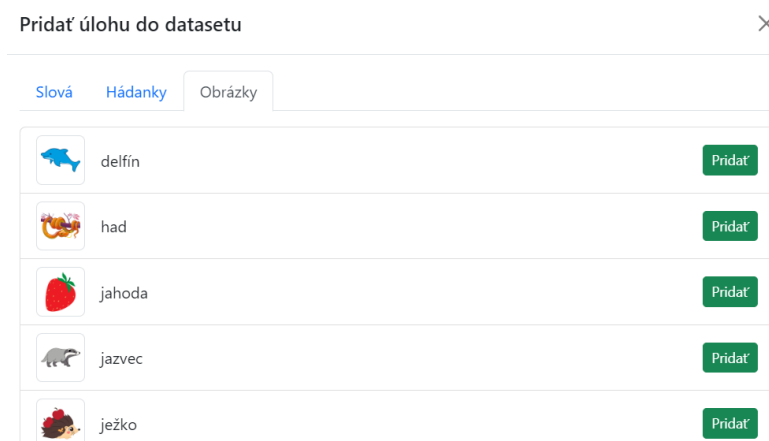
3.11.3. Pridávanie nových úloh do datasetu

Pridávanie nových úloh bolo realizované špeciálnym spôsobom, kde bolo potrebné najprv zobraziť úlohy, ktoré nie sú súčasťou datasetu (nenachádzajú sa v prepojovacích tabuľkách pri konkrétnom datasete). Kliknutím na tlačidlo „Pridať“ súbor 2 odošle AJAX dopyt s akciou `get_missing_tasks`. Súbor 3 v rovnomennej funkcii vykoná netradičný SQL dopyt `SELECT`, kde podmienkou je negovaný `SELECT` dopyt, ktorý zisťuje či v prepojovacej tabuľke sa nachádza záznam s identifikátorom konkrétnej úlohy a datasetu. Ak sa taký záznam nájde, tak úloha nie je vybraná do výpisu.

```
// Obrázky, ktoré ešte nie sú v datasete
$picturesQuery = "SELECT picture_id, picture_name, folder FROM pictures
WHERE picture_id NOT IN (SELECT picture_id FROM dataset_pictures WHERE dataset_id = ?)";
```

Obr. 66 - Využitie dvojitého SELECT dopytu

Úlohy boli oddelené podľa kategórie, teda administrátor má nad pridávaním väčšiu kontrolu. Pridávanie je už realizované podobným princípom, kde kliknutím na tlačidlo AJAX vyšle požadovanú akciu na pridanie záznamu do prepojovacej tabuľky pomocou SQL príkazu `INSERT`. Po pridaní sa úloha zo zoznamu odstráni a následne sa objaví medzi položkami zoznamu.



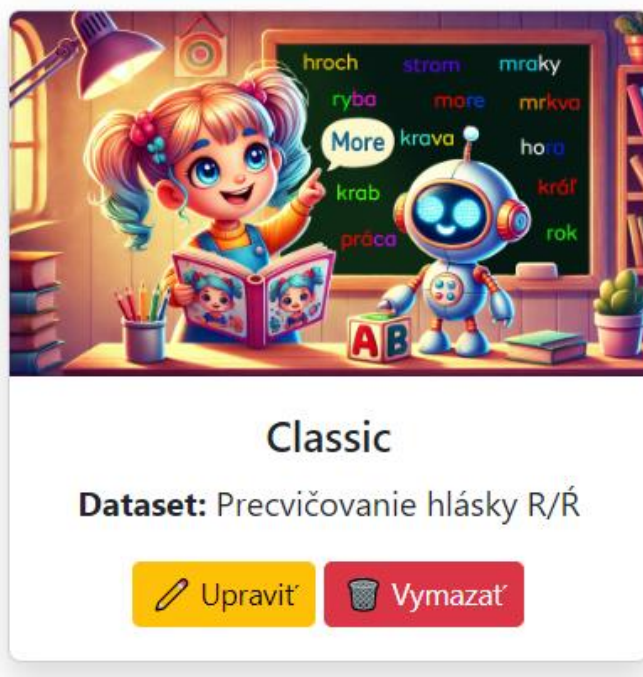
Obr. 67 - Pridanie úlohy do datasetu

3.12. Implementácia hier

Po vytvorení implementácie datasetov bolo možné pristúpiť ku možnosti vytvorenia hry. Funkcia vyžadovala najviac práce so súbormi a taktiež aj po funkčnej stránke.

3.12.1. Zobrazenie hier

Na začiatok bola vytvorená najjednoduchšia funkcia, ktorá bola vytvorená už niekoľkokrát v práci. Do súboru 1 bol pridaný element <div>, do ktorého sa budú načítavať hry. Súbor 2 pri načítavaní použije funkciu loadGames(), ktorá využije AJAX dopyt s akciou get_games. Odpoveďou na túto akciu je SQL dopyt SELECT, uložený v súbore 3, ktorý vyhledá všetky záznamy v tabuľke games. Výsledky sú následne vypísané pomocou funkcie append(). Každá hra obsahuje názov, dataset, obrázok a tlačidlá na odstránenie, úpravu a pridelenie prístupu do hry, ak hra je nastavená ako neverejná.



Obr. 68 - Výpis hry

3.12.2. Pridávanie nových hier

Pridávanie je realizované pomocou tlačidla „Pridať hru“, ktoré bolo nastavené, aby vypísalo vyskakovací formulár pri jeho aktivácii. Do formulára boli pridané vstupné elementy pre zadávanie mena hry, voľby datasetu (zoznam vytvorených datasetov sa získava zo súboru 3 pomocou SQL dopytu SELECT). Ako ďalšie boli vytvorené vstupné elementy pre vloženie obrázkov pozadia, textúry hráča 1 a hráča 2 vo formáte PNG. Pri vložení obrázka bola vytvorená v súbore 2 funkcia previewImage(), ktorá reaguje na pridanie obrázka do formulára a vykreslí obrázok, aby používateľ vedel, čo vložil do formulára.

Ďalší vstupný element slúži na vloženie návrhu mapy vo formáte JSON. Pri možnosti pridania mapy boli implementované dve možnosti pridania: pomocou vopred pripraveného súboru alebo priamym vytvorením vo formulári. Priame vytvorenie vo formulári bolo realizované pomocou tlačidla „Vytvoriť mapu“. Súbor 2 následne vykreslí nové vyskakovacie okno, do ktorého vykreslí mriežku o rozmeroch 16x9 pomocou funkcie generateGrid(). Na Obr. 69 môžeme vidieť editor mapy.



Obr. 69 - Editor mapy

Následne bola implementovaná funkcia, ktorá umožní reagovať na kliknutia a vytvoriť herné pole podľa požiadaviek používateľa. Kliknutím na element získa poradové číslo, ktoré bude kľúčové pre správnu funkciu vykreslenia hry v reálnej prevádzke. Ak sa používateľ pomýlil a chce vymazať konkrétny blok, môže tak urobiť opätovným kliknutím na tento blok. Následne sa z bloku odstráni poradové číslo. Ak sa náhodou vymazal blok, ktorého číslo nebolo najväčším, využije sa funkcia renumberGrid(), ktorá očísľuje všetky aktívne bloky od začiatku, aby sa predišlo nekonzistentnosti. Bolo implementované aj globálne vymazanie celej mriežky pomocou tlačidla „Vymazať“.

Bola taktiež vytvorená možnosť vloženia už pripraveného súboru, ktorý sa pomocou funkcie loadMap() načíta a automaticky zobrazí v editore. Funkcia bola vytvorená z dôvodu, ak by niekto

chcel editovať vytvorenú mapu, aby ju nemusel vytvárať od začiatku. Poslednou funkciou bola možnosť stiahnutia súboru do počítača, ktorá pri potvrdení „Uložiť“ okrem pridania nového súboru do formulára, automaticky vytvorí stiahnuteľný súbor, ktorý je možné uložiť do počítača.

 Editor mapy
 ✕

																	
	1	2	3	4		18	19	20		34	35	36					
				5		17		21		33		37					
				6		16		22		32		38					
				7		15		23		31		39	40	41			
				8		14		24		30							
				9		13		25		29							
				10	11	12		26	27	28							
																	

Vybrať súbor

☐ Stiahnuť mapu do počítača

 Vymazať

 Uložiť

Obr. 70 - Vytvorená mapa

Posledným vstupným elementom vo formulári je možnosť voľby verejnosti hry. Uložením hry sa všetky hodnoty v súbore 2 vložia do premennej formData, ktorá je následne odoslaná cez AJAX dopyt spolu s akciou create_game na server. V súbore 3 sa následne vykonávajú akcie ukladania súborov na server a uloženia hodnôt do databázy, kde ukladanie súborov prebieha v troch fázach. Prvou fázou je overenie, či hra pod rovnakým názvom neexistuje, ak existuje nahrávanie sa preruší. Následne vytvorí nový priečinok s názvom hry pomocou funkcie mkdir(), ktorý ešte predtým skontroluje pomocou funkcie is_dir(), či priečinok s týmto názvom neexistuje. Druhou fázou je premenovanie vstupných súborov na vopred definované hodnoty: „background.png“, „character.png“, „character2.png“ a „map.json“. Týmto bolo docielené zachovanie korektnosti načítavania pri samotnej hre, ktorá potrebuje pracovať s presnými názvami. Poslednou fázou je samotné uloženie súbor do nového priečinka pomocou funkcie move_uploaded_file(). Do databázy pomocou SQL príkazu vložíme názov hry, názov priečinka, identifikátor datasetu a verejnosť hry.

3.12.3. Úprava a odstránenie hry

Úprava hry obsahuje veľmi podobnú funkcionálnosť ako pridávanie súborom. Najväčším rozdielom je vopred vyplnení formulár pre názov, dataset, mapu a verejnosť hry. Súbor vo formulári nie sú povinné, kde pri ich nenačítaní sa zachovávajú pôvodné súbory. Pri otvorení editora mapy je mapa automaticky načítaná, čím je umožnená okamžitá editácia. Potvrdením formulára sú následne hodnoty opäť uložené do premennej formData a poslané cez AJAX na server s akciou update_game. Súbor 3 následne vykoná podobné operácie ako v odseku 3.12.2, kde bola pridaná nová funkcionálnosť pre kopírovanie a odstránenie priečinkov. Ak sa používateľ rozhodol upraviť meno hry, automaticky sa vykoná prekopírovanie všetkých aktuálnych, ale aj nových súborov do nového priečinka pomocou rekurzívnej funkcie copyFolder() a pôvodný priečinok odstráni pomocou druhej rekurzívnej funkcie deleteFolder(). Následne postup je rovnaký ako pri vytváraní hry, kde posledným rozdielom je iba zmena SQL príkazu na UPDATE.

```
// Rekurzívna funkcia na vymazanie priečinka a obsahu
3 usages
function deleteFolder($folderPath)
{
    foreach (scandir($folderPath) as $item) {
        if ($item == "." || $item == "..") continue;
        $itemPath = $folderPath . DIRECTORY_SEPARATOR . $item;
        is_dir($itemPath) ? deleteFolder($itemPath) : unlink($itemPath);
    }
    rmdir($folderPath);
}
```

Obr. 71 - Rekurzívna funkcia odstránenia priečinka

```
2 usages
function copyFolder($src, $dst)
{
    if (!is_dir($src)) return;
    foreach (scandir($src) as $item) {
        if ($item == "." || $item == "..") continue;
        $srcFile = $src . DIRECTORY_SEPARATOR . $item;
        $dstFile = $dst . DIRECTORY_SEPARATOR . $item;
        if (is_dir($srcFile)) {
            mkdir($dstFile, permissions: 0777, recursive: true);
            copyFolder($srcFile, $dstFile);
        } else {
            copy($srcFile, $dstFile);
        }
    }
}
```

Obr. 72 - Rekurzívna funkcia kopírovania priečinka

3.12.4. Priradovanie používateľov

Ako bolo vysvetlené v bode 3.12.1, hráč ktoré boli nastavené ako neverejné je možné priradovať prístup používateľom. Kliknutím na tlačidlo „Spravovať prístup“ sa zobrazí pomocou súboru 2 vyskakovacie okno, ktoré zobrazuje dva výstupy: aktuálni používatelia a potenciálni používatelia. Oba výstupy sú vygenerované kliknutím na kategóriu. Súbor 2 vyžiada zoznam hráčov zo servera pre zvolenú hru pomocou akcie `get_game_users`. Súbor 3 následne vykoná dva SQL dopyty `SELECT`, prvý pre získanie aktuálnych používateľov cez `JOIN SELECT` na prepojavacu tabuľku `user_games` a druhý pre získanie potenciálnych používateľov cez `SELECT`, kde podmienka je negovaný `SELECT` dopyt.

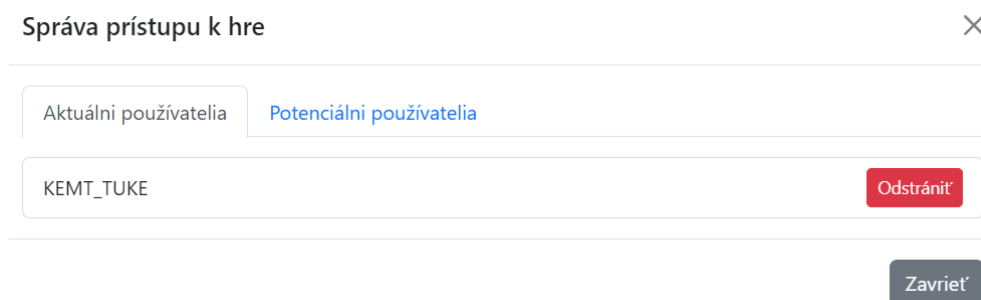
```
// Získanie užívateľov s prístupom k hre
$query = "SELECT u.id, u.username FROM users u
          JOIN user_games ug ON u.id = ug.user_id
          WHERE ug.game_id = ?";

// Získanie užívateľov, ktorí ešte nemajú prístup k hre
$query = "SELECT id, username FROM users
          WHERE id NOT IN (SELECT user_id FROM user_games WHERE game_id = ?)";
```

Obr. 73 - `SELECT` dopyty pre aktuálnych/potenciálnych používateľov

AJAX odpoveď následne vyzerá ako dve premenné obsahujúce zoznam aktuálnych a potenciálnych používateľov, ktoré sa zobrazia.

Aktuálnych používateľov je možné odstrániť a potenciálnych používateľov je možné pridať do hry. Funkcionalita sa vykonáva v súbore 2, kde určí o akú akciu ide: `add_user_to_game`, `delete_user_from_game`, získa sa identifikátor elementu a používateľa a vykoná AJAX dopyt, ktorý upozorní server o akú akciu ide. Server následovne v súbore 3 určí správnu akciu a vykoná pridanie používateľa do prepojavacej tabuľky pomocou SQL príkazu `INSERT` alebo ho odstráni z prepojavacej tabuľky pomocou `DELETE`. Odpoveďou na AJAX je úspešnosť akcie, po ktorej sa následne odstráni používateľ zo zoznamu a pridá sa do druhého zoznamu.



Obr. 74 - Aktuálni používatelia

3.13. Implementácia správy používateľov

Finálnou implementáciou do administrátorského prostredia bola správa používateľov. Funkcionalita podstránky je jednoduchá, kde neboli použité žiadne nové funkcie. V súbore 1 bol vytvorený jednoduchý výpis používateľov, ktorý je realizovaný priamo pomocou SQL dopytu SELECT, ktorý kontaktuje databázu. Následne sa všetci používatelia vypíšu. Každému používateľovi bolo pridané tlačidlo na zobrazenie účtu a možnosť odstránenia. Odstránenie účtu je umožnené iba administrátorom (administrátor nemôže vymazať iných administrátorov) a je realizované pomocou AJAX požiadavky v súbore 2, ktorý kontaktuje súbor 3, kde funkcia vykoná jediný SQL príkaz DELETE. Po odstránení z databázy sa používateľ vymaže aj z výpisu. Na záver bola implementovaná ešte jedna funkcia, ktorá umožní vyhľadávanie používateľov. Očakávame, že pri väčšej prevádzke môže byť vyhľadávanie komplikované, preto je umožnené vyhľadávanie pomocou prihlasovacích mien. Vyhľadávanie je realizované pomocou funkcie v súbore 2, ktorá reaguje na vstupné pole označené identifikátorom #search-user. Pri pridávaní nových znakov sa kontroluje či reťazec znakov zodpovedá niektorému prihlasovaciemu menu a zobrazí iba účty, ktoré zodpovedajú reťazcu.

```
// ● Vyhľadávanie používateľov v reálnom čase
$("#search-user").on("keyup", function () :void {
    let searchTerm = $(this).val().toLowerCase();
    $(".user-card").each(function () :void {
        let username = $(this).data("username").toLowerCase();
        $(this).toggle(username.includes(searchTerm));
    });
});
```

Obr. 75 - Funkcia vyhľadávania používateľov

KEMT

Používateľ: KEMT_TUKE 🌳

Rola: root

Meno: KEMT TUKE

Vek: 56 rokov

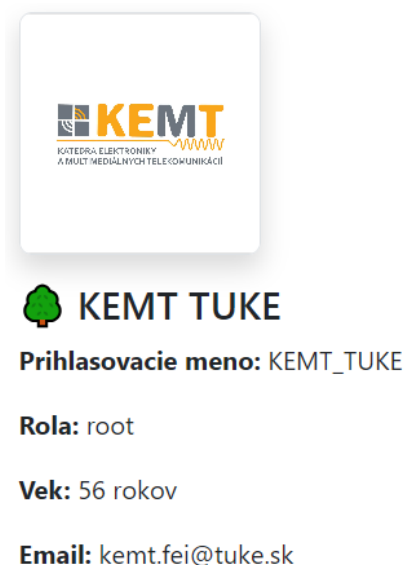
Email: kemt.fei@tuke.sk

 **Zobraziť**

Obr. 76 - Výpis používateľa

3.14. Zobrazenie používateľského účtu

Používateľské účty si môžu zobraziť všetci administrátori. Bežní používatelia môžu vidieť iba svoj účet. Na začiatok bolo vytvorené zobrazenie samotného profilu používateľa. Zobrazenie pozostáva z profilovej fotografie, používateľského mena, mena, priezviska, e-mailovej adresy a role. Všetky údaje sú načítane priamo v súbore 1 cez SQL dopyt SELECT v tabuľke users. Ak zobrazený používateľ je vlastníkom účtu, tak sú mu zobrazené tlačidlá na zmenu údajov a hesla. Zmena údajov bola realizovaná rovnakým formulárom ako pre registráciu s výnimkou zmeny prihlasovacieho mena a role. Vstupné hodnoty sú spracované súborom 2 do premennej formData, ktoré sú odoslané pomocou AJAX-u na server. Súbor 3 následne vykoná overenie novej e-mailovej adresy, uloženie nového obrázka a úpravu v databáze pomocou SQL príkazu UPDATE. Zmena hesla je podobným procesom, ale v tomto prípade bol vytvorený formulár na zadanie starého a nového hesla. Súbor 3 následne vykoná overenie starého hesla, či zodpovedá odtlačku v databáze. Ak sa odtlačky zhodujú, novému heslu vytvorí sa odtlačok a uloží sa do databázy pomocou SQL príkazu UPDATE.



Obr. 77 - Zobrazenie používateľa

Ďalšia funkcionálnosť, ktorá bola implementovaná na podstránku je zobrazenie výsledkov hier. Výsledky hier sa vyhľadávajú rovnako ako samotný používateľ SQL dopytom SELECT z tabuľky game_results. Samotný výpis pozostáva z mena hry, dátumu hry, počtu bodov a herného režimu. Podrobnejšie výsledky je možné vypísať pomocou tlačidla „Zobraziť odpovede“. Kliknutím na tlačidlo súbor 2, pošle AJAX dopyt s akciou get_answers. Odpoveďou na dopyt sú vyhľadované odpovede, ktoré prislúchajú ku danej hre, ktorá je vyjadrená identifikátorom hry. Následne je vykreslené vyskakovacie okno, ktoré obsahuje podrobné výsledky danej hry.

Detail odpovedí ×

Otázka	Odpoveď	Správna odpoveď	Body	
Odpovedz na hádanku.	dver	vietor	0	
Odpovedz na hádanku.	zebra	zebra	2	
Dokonči názov obrázka.	hrozno	hrozno	1	
Označ obrázok, ktorý tu nepatrí.	ruka	prst	0	
Označ obrázok, ktorý tu nepatrí.	dcéra	dcéra	3	
Dokonči názov obrázka.	kanárik	kanárik	1	
Označ obrázok, ktorý tu nepatrí.	dcéra	dcéra	3	

Zavrieť

Obr. 78 - Zobrazenie podrobných výsledkov

3.15. Implementácia úvodnej stránky

Po implementácii celej administrátorskej časti bolo možné vytvoriť hlavnú stránku, ktorá bude slúžiť ako vstup do platformy. Väčšina funkcionality bola už vysvetľovaná v predošlých odsekoch. Na úvodnej stránke bola vytvorená navigačná lišta, opis platformy, dostupné hry, sekciu o autoroch a kontaktné údaje. Okrem výpisu dostupných hier sú všetky ostatné položky statické, teda fixne vložené. Výpis dostupných hier je dynamická časť, ktorá pomocou SQL dopytu SELECT vyhľadá všetky verejné hry. Ak je používateľ prihlásený a má pridelené špecifické hry, pomocou špecifického SQL dopytu SELECT JOIN sa nájdu hry, ktoré sú prepojené s hráčom cez prepojovaciu tabuľku user_games. Výpisy hier následne umožňujú presmerovanie na samotné vytvorené hry.

```
$assignedGamesQuery = "
SELECT DISTINCT g.id, g.name, g.folder, d.dataset_name AS category
FROM games g
JOIN user_games ug ON g.id = ug.game_id
LEFT JOIN datasets d ON g.dataset = d.dataset_id
WHERE ug.user_id = ? AND g.public = 0
";
```

Obr. 79 - SQL dopyt pre výpis špecificky pridelených hier



Obr. 80 - Výpis verejných hier

3.16. Implementácia hry

Poslednou časťou, ktorá bola vytvorená do platformy je samotná hra. Na vytvorenie hernej logiky bola potrebná funkčná databáza a framework Phaser 3, ktorý uľahčil našu prácu pri tvorbe.

3.16.1. Predpríprava hodnôt z databázy

Na začiatok bolo vložené do súboru game.php overovanie prihlásenia, ktoré bude potrebné na konci hry. Ako ďalšiu hodnotu bolo potrebné získať identifikátor danej hry, ktorý bude potrebný pre správne uloženie výsledkov hry. Poslúžil nám na to SQL dopyt `SELECT id FROM games WHERE folder = $game`. Do premennej `$game` a `$mode` boli na začiatku načítané hodnoty, ktoré boli odoslané z úvodnej stránky, kliknutím na požadovanú hru. Následne sa vytvorila pomocou týchto premien aj adresné cesty, ktoré budú použité pre načítanie súborov.

```
$game = isset($_GET['game']) ? $_GET['game'] : "classic";
$mode = isset($_GET['mode']) ? $_GET['mode'] : "single";
$texturePath = "assets/$game/";
$mapPath = "assets/$game/map.json";
$mapData = file_exists($mapPath) ? file_get_contents($mapPath) : "{}";
```

Obr. 81 - Načítanie adries pre priečinkov hry

3.16.2. Importovanie frameworku Phaser 3 a inicializácia globálnych premien

Framework Phaser 3 bol stiahnutý zo stránky Phaser 3⁽¹⁾, kde bola pomocou HTML tagu <script> do hlavičky súboru vložená URL adresa balíka. Použili sme verziu 3.55.2, ktorá bola dostupná v októbri 2024 (aktuálna verzia frameworku je 3.88.2, ktorá je obsiahnutá v súbore, ale je deaktivovaná pomocou komentára). Týmto spôsobom nie je potrebné sťahovať súbory zo samotnej stránky, ale sú vždy dostupné na URL adrese to znamená, že hra vyžaduje na správne fungovanie prístup k internetu.

```
<script src="https://cdn.jsdelivr.net/npm/phaser@3.55.2/dist/phaser.min.js"></script>
<!--<script src="https://cdn.jsdelivr.net/npm/phaser@v3.88.2/dist/phaser.min.js"></script-->
```

Obr. 82 - Importované frameworky Phaser 3

Ďalším krokom bola inicializácia globálnych premien, ktoré sú potrebné pre správny chod hry. Bolo využitých 46 globálnych premien, ktoré umožňujú uloženie rôznych typov údajov (systémové, načítané herné pole, súradnice hráčov, skóre, ...).

```
const game = new Phaser.Game(config);
const urlParams = new URLSearchParams(window.location.search);
const mode = urlParams.get('mode') || 'single';
let snakeShape = <?= $mapData ?>.snakeShape;
let player1; // Premenná pre P1
let player1won = false;
let player1turn = true;
let outline; // Tieň pre P1
let player2; // Premenná pre P2
let player2won = false;
let player2turn = false;
let outline2;
```

Obr. 83 - Ukážka globálnych premien

3.16.3. Funkcia preload()

Funkcia preload() je prvou základnou funkciou používaná v Phaser 3. Jej úlohou je načítanie všetkých potrebných súborov a hodnôt, ktoré budú používané v nasledujúcich funkciách. Bola využitá funkcia na načítanie textúr pre pozadie, modely postáv pre hráčov, medailu a rôzne audio nahrávky, ktoré sú súčasťou hry staticky. Súbory sa načítavajú zo servera, preto bol pridaný indikátor načítavania, ktorý hráčom oznamuje progres načítania. Zvolili sme toto riešenie pre ľudí, ktorí nemajú rýchle internetové pripojenie, aby nevznikli pochybnosti o správnej funkcionalite hry.

¹ Odkaz na stránku: <https://phaser.io/>

```

let texturePath = "<?=$texturePath ?>";
this.load.image('player1', texturePath + 'character.png');
this.load.image('player2', texturePath + 'character2.png');
this.load.image('background', texturePath + 'background.png');
this.load.image('medal', 'assets/system_assets/images/medal.png');

this.load.audio('moveSound', 'assets/system_assets/SFX/move.mp3');
this.load.audio('backgroundMusic', 'assets/system_assets/SFX/background_music.mp3');
this.load.audio('correctAnswer', 'assets/system_assets/SFX/correct_answer.mp3');
this.load.audio('applause', 'assets/system_assets/SFX/applause.mp3');
this.load.audio('wrongAnswer', 'assets/system_assets/SFX/wrong_answer.mp3');
this.load.audio('victoryFanfare', 'assets/system_assets/SFX/victory_fanfare.mp3');

```

Obr. 84 - Načítanie textúr a audio nahrávok

3.16.4. Funkcia create()

Po inicializačnej funkcii preload() sa vykonáva funkcia create(). Úlohou funkcie je vytvoriť všetky elementy, ktorá sa následne zobrazia na obrazovke hráča. Na začiatku bola vytvorená premenná scény, ktorá bude slúžiť ako hlavný element, kde sa budú ukladať všetky ďalšie elementy. Následne bol implementovaný obrázok a hudba pozadia. Ako ďalší bol vytvorený výpis herného poľa. Pozostáva z dvoch cyklov, ktoré prehľadáva dvojrozmerné pole snakeShape, kde sa nachádza načítané pole zo súboru map.json. Každý blok herného poľa je vygenerovaný o veľkosti 100 x 100 pixelov, kde sa mení farba v cykle: modrá, zelená, žltá, červená. Každá farba následne bude hrať úlohu pri výbere úlohy. Ak je hodnota herného políčka rovná 1, uložia sa do premenných hráča 1 hodnoty súradníc umiestnenia na obrazovke.

```

// Vykreslenie hracieho poľa
const cellSize = 100; // Veľkosť jednej bunky
let color;
for (let i = 1; i < snakeShape.length - 1; i++) {
  gridColors[i] = [];
  for (let j = 1; j < snakeShape[i].length - 1; j++) {
    if (snakeShape[i][j] > 0) {
      if (snakeShape[i][j] % 4 === 1) color = 0x0ddfe0;
      else if (snakeShape[i][j] % 4 === 2) color = 0x00ff00;
      else if (snakeShape[i][j] % 4 === 3) color = 0xffff00;
      else color = 0xff0000;
      let cell = this.add.rectangle((j * cellSize) - 100, (i * cellSize) - 100, cellSize, cellSize, color);
      cell.setOrigin(0);
      cell.setStrokeStyle(2, 0x000000);
      cell.setAlpha(0.9);
      gridColors[i][j] = color;
    }
  }
}

```

Obr. 85 - Cyklus pre vykreslenie herného poľa

Po vykreslení herného poľa sa následne vykreslí medaila, ktorá symbolizuje koniec hry na herné políčko, ktorého číslo je najvyššie zo všetkých. Následne je vykreslená samotná figúrka hráča 1 a jeho tieň. Ak sa jedná o režim dvoch hráčov tak sa vykreslí aj hráč 2. Súradnice hráčov sú uložené

v dvoch rovinách: grafickej a sieťovej. Grafická rovina obsahuje reálne súradnice, pomocou ktorých sa pohybuje herná figúrka. Sieťová rovina reprezentuje súradnice herného poľa, ktoré je uložené v premennej snakeShape a umožňuje tak funkčný a matematický pohyb po hernom poli, tak aby nenastal problém s funkcionalitou.

```
// vykreslenie hráča/ov
outline = this.add.sprite(50 + (100 * (player1_y - 1)), 50 + (100 * (player1_x - 1)), 'player1');
outline.setDisplaySize(90, 100); // Väčší ako hlavný sprite
outline.setTint(0x000000); // Čierny obrys
outline.setDepth(0); // Pod hráčom
outline.setOrigin(0.5);

player1 = this.add.sprite(50 + (100 * (player1_y - 1)), 50 + (100 * (player1_x - 1)), 'player1');
player1.setDisplaySize(80, 100);
player1.setOrigin(0.5);
```

Obr. 86 - Vykreslenie hráča

Ako posledné elementy sa vygenerujú rôzne kontajnery pre výpis úvodných pokynov, víťazného upozornenia alebo kontajnery, do ktorých sa budú zobrazovať úlohy alebo zobrazovať aktualizovaná hodnota skóre. Pre zachovanie jednoduchosti nebudeme uvádzať všetky kontajnery podrobne.

```
background2.setStrokeStyle(2, 0x000000);
messageText = this.add.text(0, 180, 'Rozpoznané slovo je:', {
  fill: '#ffffff', fontSize: '30px', fontStyle: 'bold',
  fontFamily: 'Arial', stroke: '#000000', strokeThickness: 8, align: 'center',
  wordWrap: {width: 800, useAdvancedWrap: true}
});
messageText.setOrigin(0.5);
gamecontainer.add(background2);
gamecontainer.add(messageText);
```

Obr. 87 - Ukážka vloženia textu do kontajnera

Na záver bolo importované samotné Web Speech API, ktoré zabezpečí hlavnú funkcionalitu hry, teda rozpoznanie reči. Bolo nastavené rozpoznávanie jazyka na hodnotu „sk-SK“, podľa ktorej vie algoritmus definovať slovenský jazyk. Následne spracuje reč, kde odpoveďou je rozpoznané slovo zapísané malými písmenami. Na Obr. 88 je možné vidieť kód, ktorým bolo importované Web Speech API.

```
// Nastavenie rozpoznávania reči (Web Speech API)
if ('webkitSpeechRecognition' in window || 'SpeechRecognition' in window) {
  const SpeechRecognition = window.SpeechRecognition || window.webkitSpeechRecognition;
  speechRecognition = new SpeechRecognition();
  speechRecognition.lang = 'sk-SK'; // Jazyk pre rozpoznávanie reči
  speechRecognition.interimResults = false;

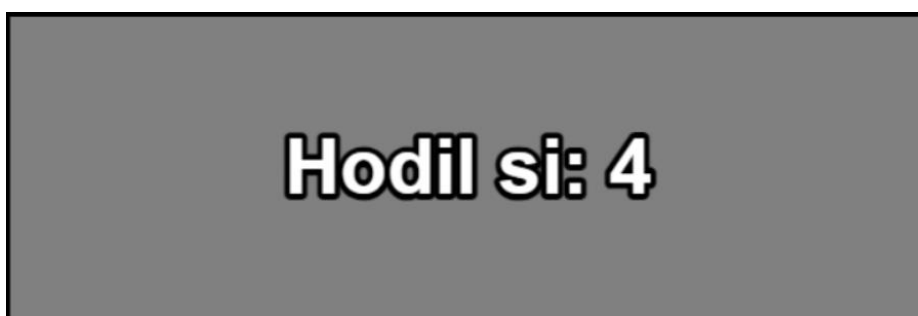
  speechRecognition.onresult = (event) => {
    const command = event.results[0][0].transcript.toLowerCase();
    handleSpeechInput(command, scene);
  };

  speechRecognition.onerror = (event) => {
    console.error('Chyba rozpoznávania reči:', event.error);
  };
} else {
  console.error('Web Speech API nie je podporované vo vašom prehliadači.');
```

Obr. 88 - Importovanie Web Speech API

3.16.5. Funkcia rollDice()

Funkcia rollDice() zabezpečuje výpočet počtu krokov, o ktoré sa môže hráč posunúť. Funkcia bola nastavená rôzne pre režim jedného a dvoch hráčov. Pri režime jedného hráča bol vytvorený jednoduchý generátor náhodných čísel, ktorý je vopred pripravený vo frameworku pomocou príkazu Phaser.Math.Between(1, 6), čím bola docielená simulácia hodu kockou. V režime dvoch hráčov bol zvolený model, ktorý je viac závislý na správnej odpovedi hráča. Na začiatku sa hráči posunú vždy o 1 políčko dopredu. Následne správnou odpoveďou si zabezpečujú pohyb po hernom poli. Po určení hodnoty sa následne hodnota vypíše pomocou funkcie tweens.add(), ktorá zabezpečí plynulé zobrazenie kontajneru s textom. Po vykonaní animácie sa zavolá funkcia moveForward(), do ktorej je uložená hodnota počtu posunutí na hernom poli.



Obr. 89 - Zobrazenie hodu kockou

3.16.6. Funkcia moveForward()

Funkcia zabezpečuje pohyb po hernom poli. Na začiatku sa overí, či hodnota počtu krokov sa nerovná 0. Následne sa vykoná pohyb pomocou vnorenej funkcie moveSteps(), ktorá zabezpečuje pohyb po jednom políčku. Funkcia overí na sieťovej mape v premennej snakeShape, kde sa nachádza ďalšie vyššie číslo vzhľadom na aktuálnu pozíciu hráča. Podľa umiestnenia ďalšieho

políčka, na ktorom je vyššia hodnota čísla sa nastaví hodnota deltaX a deltaY. Hodnoty deltaX a Y môžu nadobúdať hodnoty -1, 0 a 1, ktoré následne určia, akým smerom sa má vykonať pohyb:

- **deltaY == -1** -> doľava,
- **deltaY == 1** -> doprava,
- **deltaX == -1** -> hore,
- **deltaX == 1** -> dole.

Vždy jedna z týchto hodnôt musí byť nulová aby sa zabezpečilo, že sa hráč nebude pohybovať diagonálnym pohybom.

```
if (currentPlayer_y + 1 < snakeShape[0].length && snakeShape[currentPlayer_x][currentPlayer_y + 1] == currentPlayer_grid + 1) {
    deltaY = 1; // Posun doprava
    moved = true;
} else if (currentPlayer_y - 1 >= 0 && snakeShape[currentPlayer_x][currentPlayer_y - 1] == currentPlayer_grid + 1) {
    deltaY = -1; // Posun doľava
    moved = true;
} else if (currentPlayer_x + 1 < snakeShape.length && snakeShape[currentPlayer_x + 1][currentPlayer_y] == currentPlayer_grid + 1) {
    deltaX = 1; // Posun dole
    moved = true;
} else if (currentPlayer_x - 1 >= 0 && snakeShape[currentPlayer_x - 1][currentPlayer_y] == currentPlayer_grid + 1) {
    deltaX = -1; // Posun hore
    moved = true;
}
```

Obr. 90 - Určenie polohy pohybu

Po určení polohy ďalšieho herného políčka sa vykoná samotný pohyb zmenením súradníc hráča na obrazovke a taktiež aj v rámci sieťovej polohy. Na konci pohybu sa overí, či sa hráč nenachádza na poslednom políčku hry, kde hru okamžite ukončí, zahrá víťaznú fanfáru a oznámi hráčovi celkové skóre, resp. ktorý hráč vyhral, ak je aktivovaný režim dvoch hráčov.

Ak nenastal koniec hry a počet krokov je väčší ako nula, rekurzívne sa zavolá funkcia moveSteps(). Posledným volaním, kde hodnota počtu odstavajúcich krokov sa rovná 0 sa vykoná inicializácia úlohy. Určí sa aká farba sa nachádza na aktuálnom políčku, podľa ktorej sa určí úloha:

- **Modrá** – obrázková úloha,
- **Zelená** – hádanka,
- **Žltá** – jednoduchá úloha,
- **Červená** – doplnenie slova.

Po upozornení hráča sa vykoná funkcia podľa farby políčka, ktorá načíta typ úlohy. Správnu odpoveď si vloží do premennej, ktorá následne bude používaná na overenie rozpoznaného slova. Na záver sa aktivuje tlačidlo na spustenie rozpoznávania reči.

3.16.7. Funkcia loadImages()

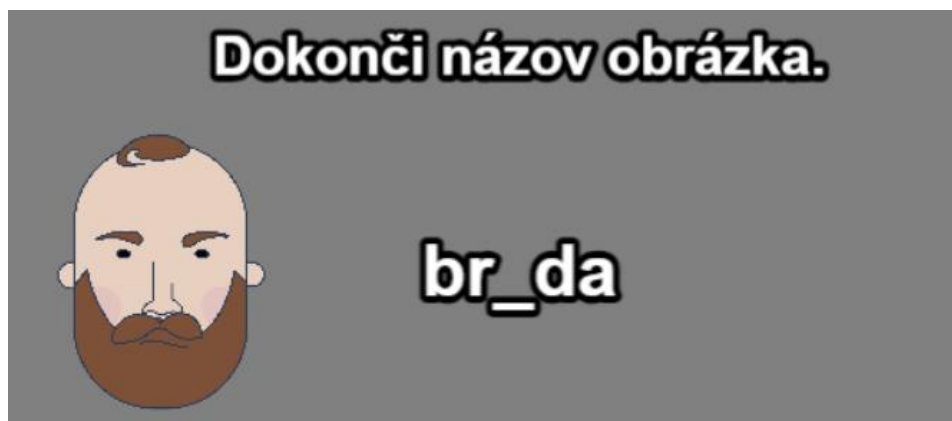
Funkcia zabezpečuje načítanie obrázkov, ak je potrebné načítať podklady pre obrázkovú úlohu a doplnenie slova. Je to asynchrónna funkcia, ktorá dokáže načítať obrázky bez potreby obnovenia stránky. Toto riešenie neobsahuje AJAX z dôvodu vytvorenia funkcie pred realizáciou administrátorskej časti. Funguje na princípe volania funkcie fetch(), ktorá umožňuje podobnú odpoveď ako AJAX. Na začiatku bola vytvorená podmienka, ktorá určí, či je potrebné načítať jeden alebo viac obrázkov. Následne sa pomocou funkcie fetch(), odošle dopyt do súborov get_images.php alebo get_image.php. Obidva súbory fungujú na podobných princípoch ako sme používali v administrátorskej časti. Súbor get_images.php obsahuje SQL dopyty SELECT, kde sa najprv získa obrázok, ktorý sa nachádza v datasete a následne vyberie tri náhodné obrázky, ktoré sa nenachádzajú v datasete a nie sú z rovnakej kategórie ako prvý obrázok. Výsledky sa odošlú späť do hry ako súbor typu JSON. Súbor get_image.php pracuje na rovnakom princípe ale využíva iba prvý SELECT dopyt.

Po načítaní obrázkov sa následne vykoná vykreslenie a vytvorenie textu úlohy, ktorý slúži ako inštrukcia pre hráča. Ak je zvolená obrázková úloha, obrázky sa náhodne premiešajú pomocou príkazu `data.sort(() => Math.random() - 0.5)`. Následne odstráni predošlé obrázky a vykreslí ich. Ako pomôcka bola vytvorená možnosť zobrazenia názvu obrázka prechodom kurzoru myši na obrázok.



Obr. 91 - Vykreslenie obrázkovej úlohy

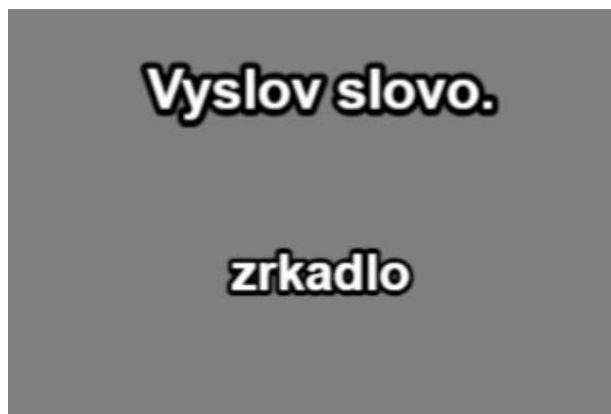
Doplnenie slova funguje na rovnakom princípe, kde zmena nastáva pri výpise inštrukcií a výpise slova. Výpis je realizovaný odstránením jedného náhodného písmena zo slova, kde písmeno je nahradené symbolom “_”. Na záver sa vráti hodnota správneho slova do funkcie moveForward(), ktorá následne uloží slovo.



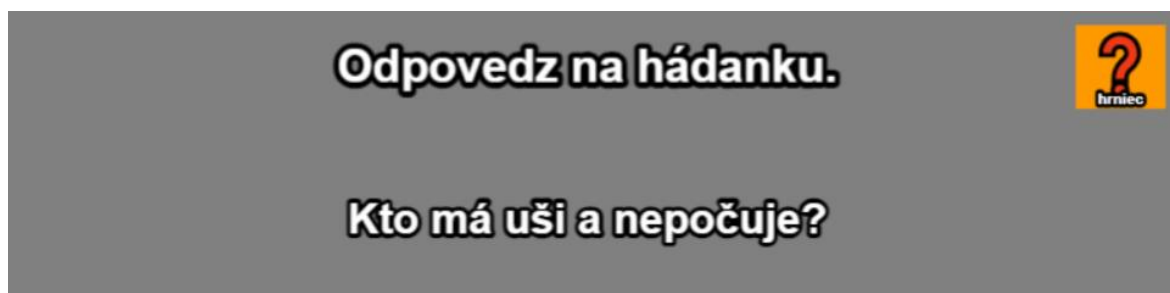
Obr. 92 - Úloha doplnenia slova

3.16.8. Funkcia loadRiddle()

Funkcia loadRiddle() umožňuje načítanie textových úloh, pod ktoré patria jednoduché úlohy a hádanky. Postup realizácie bol totožný ako pri funkcií loadImages(), kde sa na začiatku určí o akú akciu ide a podľa nej sa vykoná požiadavka na súbor get_simple_word.php alebo get_riddle.php. Oba tieto súbory obsahujú SQL dopyty SELECT pre výber úlohy z datasetu. Následne je výpis zjednodušený pretože nenačítava obrázky, ale iba textové hodnoty. Bola taktiež vytvorená pomôcka pre hádanky, kde sa po 5 sekundách zobrazí štvorec označený otáznikom, ktorý umožní vidieť správnu odpoveď pri prechode na štvorec kurzorom myši.



Obr. 93 - Zobrazenie jednoduchkej úlohy



Obr. 94 - Zobrazenie a hádanky a pomôcky

3.16.9. Funkcia handleSpeechInput()

Funkcia, ktorej najdôležitejšou úlohou je vyhodnotenie správnej odpovede. Ak hráč vyslovil slovo, ktoré sa zhoduje so správnou odpoveďou, vykonajú sa úlohy závislé na režime hry. Pri režime jedného hráča sa pri správnej odpovedi pripočíta hráčovi nasledovný počet bodov:

- **Obrázková úloha** -> 300 bodov,
- **Hádanka** -> 200 bodov,
- **Jednoduché slovo** -> 100 bodov,
- **Doplnenie slova** -> 100 bodov.

Pri režime dvoch hráčov nie sú prideľované body, ale počet krokov, ktoré môže vykonať hráč v ďalšom kole:

- **Obrázková úloha** -> 2 kroky,
- **Hádanka** -> 2 kroky,
- **Jednoduché slovo** -> 1 krok,
- **Doplnenie slova** -> 1 krok.

Ak hráč neodpovie správne nedostáva bodové ohodnotenie a neposunie sa dopredu. Správna a nesprávna odpoveď je zvýraznená zvukovou nahrávkou indikujúcu správnosť odpovede. Následne sa údaje ako názov úlohy, počet bodov, hráčova odpoveď, správna odpoveď a správnosť hráčovej odpovede vložia do poľa gameResults pomocou funkcie push(), ktoré sa použije následne pri ukladaní výsledkov do databázy. Pri režime dvoch hráčov sa ukladanie výsledkov uskutočňuje iba pri ťahu hráča 1, ktorý je v roli prihláseného používateľa. Na záver funkcie sa vykonajú operácie, ktoré pri hre dvoch hráčov nastaví premenné na ťah oponenta.

The image shows a dark grey rectangular box with white text. The text is centered and consists of two lines. The first line says "Rozpoznané slovo je: hrniec" and the second line says "Huráá hovoríš správne, dostávaš 200 bodov." The text has a slight shadow effect.

Obr. 95 - Správna odpoveď

The image shows a dark grey rectangular box with white text. The text is centered and consists of three lines. The first line says "Rozpoznané slovo je: slon", the second line says "Škoda nepovedal si správne slovo", and the third line says "Správne slovo bolo: papier". The text has a slight shadow effect.

Obr. 96 - Nesprávna odpoveď

3.16.10. Funkcia saveGameResults()

Záverečnou funkciou, ktorá bola implementovaná je funkcia na uloženie herných výsledkov do databázy. Vo funkcií na rozdiel od ostatných asynchrónnych funkcií sa využíva AJAX, z dôvodu implementácie na úplný záver projektu. Avšak pre správne fungovanie s Phaser 3 frameworkom bolo potrebné použiť tzv. Promise. Ten zabezpečí, že umožní vykonanie presmerovania na úvodnú stránku až po vykonaní uloženia do databázy. AJAX následne pošle výsledky uložené v premennej gameResults do súboru save_results.php. V súbore sa vykonávajú 3 SQL príkazy. Prvým dopytom je získanie identifikátora používateľa, ktorý sa použije pri ukladaní dát. Ako ďalší krok sa vykoná suma počtu bodov, ktoré sú súčasťou čiastkových výsledkov. Druhým SQL príkazom je INSERT, ktorý vloží do tabuľky game_results hodnoty identifikátora hry a hráča, súčtu bodov a herného režimu. Následne sa vykoná posledný SQL príkaz INSERT, ktorý do tabuľky game_answers vloží postupne všetky výsledky prepojené s tabuľkou game_results s využitím identifikátora hry. Po úspešnom uložení do databázy sa hráčovi objaví potvrdenie o uložení údajov.

Funkciou saveGameResults() sme ukončili kapitolu 3, ktorou sme si vysvetlili funkcionality platformy. V poslednej kapitole sa budeme venovať vytvoreniu demo hry a testovaniu platformy.

```
// 2 Vloženie odpovedí do game_answers
$stmt = $conn->prepare("INSERT INTO game_answers (result_id, question, answer,
        correct, points, correct_answer) VALUES (?, ?, ?, ?, ?, ?)");

foreach ($results as $result) {
    $question = isset($result['question']) ? $result['question'] : "Neznáma otázka";
    $answer = isset($result['word']) ? $result['word'] : "Neznáma odpoveď";
    $correct = isset($result['correct']) ? (int)$result['correct'] : 0;
    $points = isset($result['points']) ? $result['points'] : 0;
    $correct_answer = isset($result['answer']) ? $result['answer'] : ""; // Očakávaná odpoveď

    $stmt->bind_param("ississ", $result_id, $question, $answer, $correct, $points, $correct_answer);
    $stmt->execute();
}
```

Obr. 97 - Uloženie podrobných výsledkov

4. Finálne úpravy a testovanie

V poslednej kapitole sme sa venovali testovaniu platformy a jej nasadením do funkčnej siete.

4.1. Vytvorenie hry Classic

V platforme sme vytvorili preddefinovanú demo hru, ktorú sme nazvali „Classic“. V hre sme využili slovnú banku z bakalárskej práce, ktorá obsahovala vyše 150 slov a obrázkov. Zo slovnej a obrázkovej banky sme prevzali 33 obrázkov a 48 slov. Následne sme na stránke Najmama² vyhľadali 12 hádaniek, ktorých odpoveď obsahuje písmeno „R“. Pomocou množiny sme vytvorili dataset s názvom „Precvičovanie hlásky R/Ř“, do ktorého sme vložili všetky úlohy obsahujúce písmeno „R/Ř“.

Obrázky pre hru sme vygenerovali pomocou umelej inteligencie ChatGPT z dôvodu ukážky funkcionalít demo hry, kde obrázky obsahujú elementy z rovnakej tematiky. Zvolili sme tematiku dievčatka, ktoré učí robota vyslovovať slová. Vygenerovali sme obrázok pozadia a dvoch robotov, kde jeden má modrú a druhý červenú farbu.

Na záver sme vytvorili herné pole v súbore map.json, ktoré má tvar hada dlhého 41 políčok. Hra bude vždy dostupná pri nasadení platformy, keďže je obsiahnutá v priečinku platformy. Ak administrátor by chcel vymazať túto hru, môže tak vykonať kedykoľvek.



Obr. 98 - Vygenerovaný obrázok pozadia

² Odkaz na stránku: <https://najmama.aktuality.sk/aktivity/hadanky>

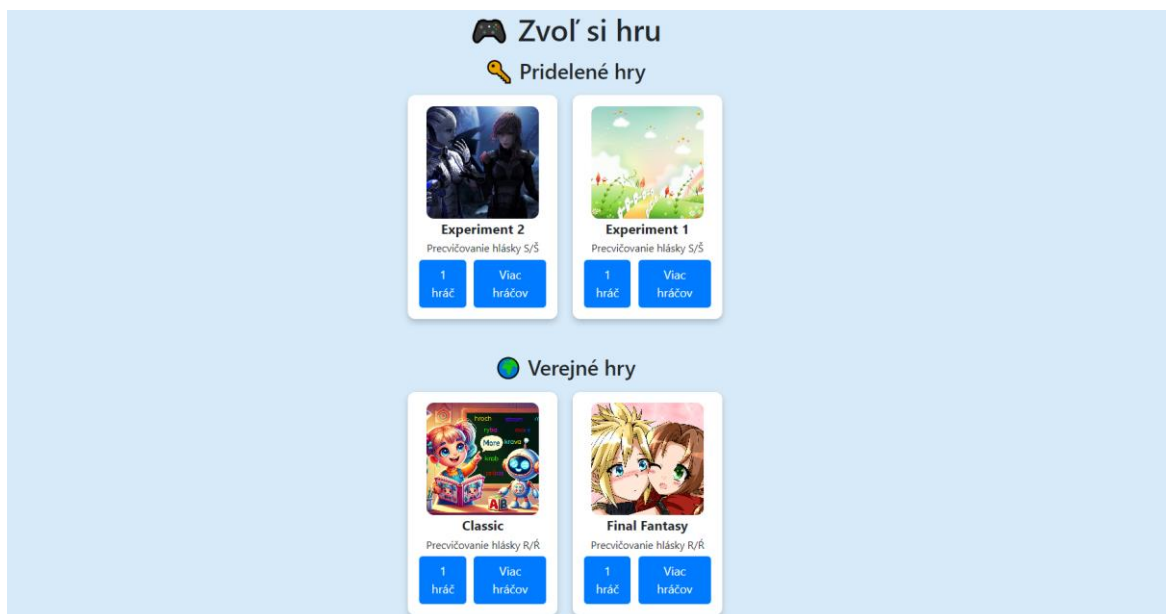
4.2. Otestovanie aplikácie

Po vytvorení platformy na lokálnom zariadení sme sa rozhodli spustiť našu aplikáciu mimo lokálnej siete s rozsahom IP adries 192.168.8.0/24. Ako prvú možnosť sme využili školskú službu Cloud TUKE, ktorá umožňuje vytvorenie virtuálneho stroja dostupného v sieti TUKE alebo cez TUKE VPN. Po podaní žiadosti nám boli pridelené prostriedky vo forme virtuálneho stroja s operačným systémom Windows 10. K tomuto stroju sme sa pripojili pomocou protokolu RDP, ktorý je štandardnou súčasťou Windows 10/11. Následne sme na server nainštalovali potrebné softvéry a nechali ho bežať.

Následne dňa 1. apríla 2025 prebehol test platformy, na ktorom sa zúčastnilo 5 osôb, ktorých úlohou bolo otestovať dizajn, funkcionality a samotnú demo hru platformy. Pri testovaní ľudia využili niekoľko typov zariadení vrátane notebookov typu ASUS TUF A15 FA507, ASUS ROG Strix GL553VD a ASUS ROG Strix G17, ktoré disponovali rozlíšením 1920x1080 a 2560x1440. Pri testoch na týchto zariadeniach nenastal žiadny problém s rozložením dizajnu platformy z dôvodu natívneho vývoja pre displeje s pomerom strán 16:9 a rozlíšením 1080p a vyššie. Samotná hra bola vykresľovaná bez problémov. Jedinou označenou chybou bolo rozloženie pozadia hry, kde pri nesprávnej veľkosti obrázka môže spôsobovať deformáciu spôsobenú natiahnutím alebo sploštením obrázka.

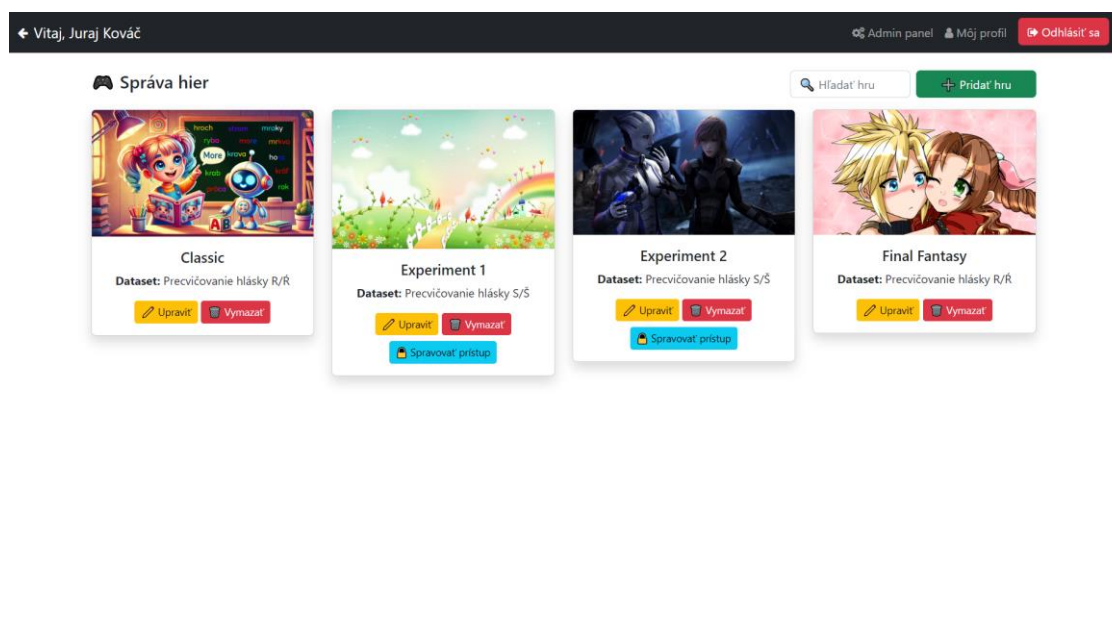
Ďalšie dva testy prebehli na mobilných zariadeniach typu Huawei P20 Lite a Samsung Galaxy S24 s rozlíšením displeja 2280x1080 a 2340x1080. Treba podotknúť, že samotná hra nebola vytvorená pre mobilné displeje z dôvodu nadmerného využívania mobilných zariadení u maloletých detí a komplikovanejšieho vývoja a úprav hry pre mobilné zariadenia. Pri otestovaní administrátorskej časti bol dizajn vhodne upravený a elementy boli dostatočne veľké. Našli sa prípady kedy niektoré tlačidlá presahovali mimo šírku displeja, ak používateľ využil vertikálnu orientáciu displeja.

Okrem dizajnovej stránky si používatelia vytvorili vlastné používateľské účty, ktorým boli pridelené administrátorské práva pre umožnenie prístupu. Následne vytvorili dva nové datasety, do ktorých sa snažili vložiť nové obrázky a aj úlohy. Pri teste neboli udelené žiadne podmienky, teda používatelia mohli vložiť čokoľvek bez ohľadu na veľkosť alebo obsah (okrem nevhodného obsahu). Zistilo sa, že zobrazenie obrázkov reaguje na pomer strán obrázka, teda vertikálne a horizontálne obrázky boli zobrazené bez deformácie. Následne používatelia vyskúšali vytvorenie vlastnej mapy, ktoré si veľmi pochvaľovali. V hre boli otestované oba režimy, kde sa používateľom páčil koncept úplnej variabilnosti hry. Niektorí používatelia na konci pridali rôzne návrhy, ktoré by mohli zlepšiť hru. Stále však musíme brať ohľad na to, že platforma je určená pre deti, teda niektoré návrhy by boli pre maloleté deti náročné a taktiež navrhovali prerobenie hry na mobilné zariadenia.



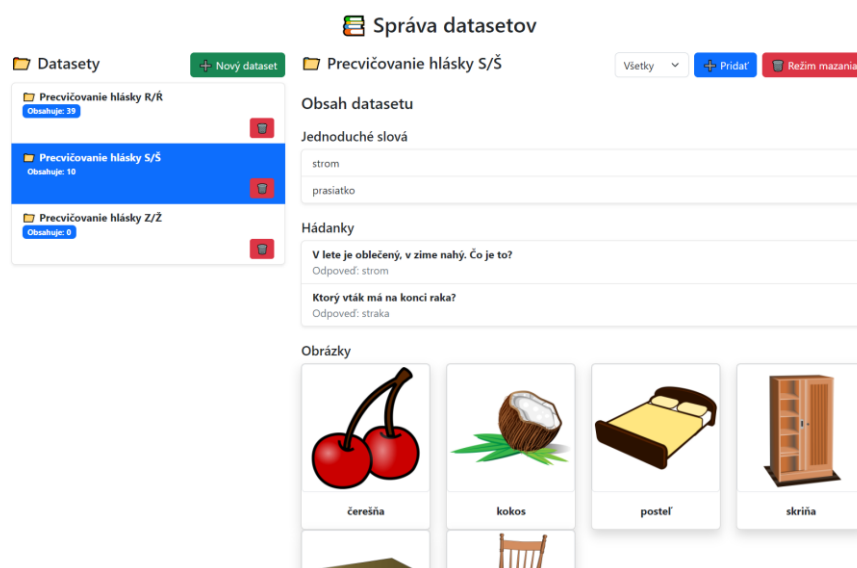
Obr. 99 - Úvodná stránka

Na Obr. 99 a Obr. 100 môžeme vidieť vytvorené hry, ktoré boli vytvorené pri testovaní. Dokopy boli vytvorené tri hry, v ktorých boli použité rôzne obrázky a vytvorené rôzne mapy.

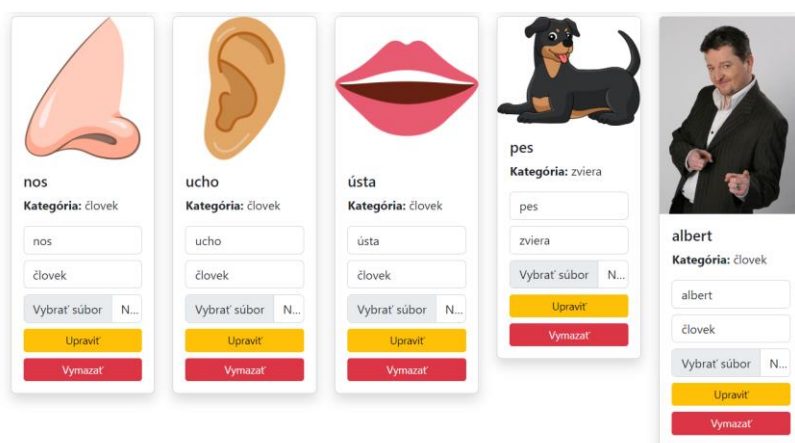


Obr. 100 - Vytvorené hry

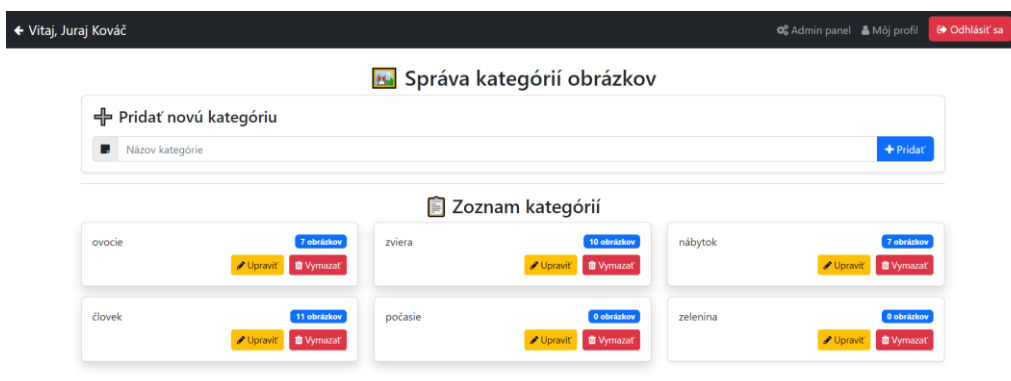
Na Obr. 101, Obr. 102 a Obr. 103 môžeme vidieť vytvorené datasety, do ktorých boli pridané rôzne obrázky, tak aby spĺňali tému datasetu, obrázkov. Taktiež môžeme vidieť pridané obrázky, ktoré majú rôznu veľkosť a pomer strán a na poslednom obrázku vidíme pokus o vytvorenie nových kategórií obrázkov.



Obr. 101 - Vytvorené datasety

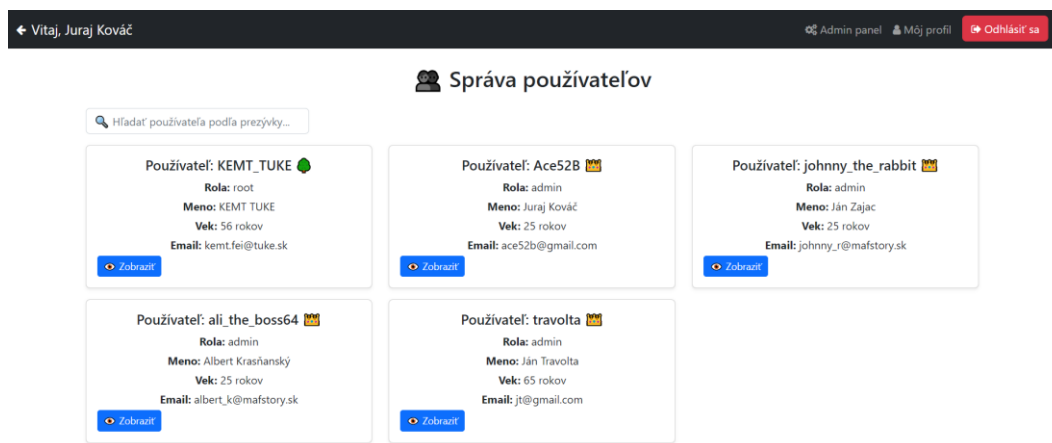


Obr. 102 - Pridané obrázky



Obr. 103 - Zoznam kategórií

Obr. 104, Obr. 105 a Obr. 106 znázorňujú vytvorených používateľov v správe administrátorov a následný priebeh testovania hry, v ktorej môžeme vidieť obrázkovú úlohu. Na záver je zobrazený finálny výsledok hráča.



Obr. 104 - Vytvorenie používateľa

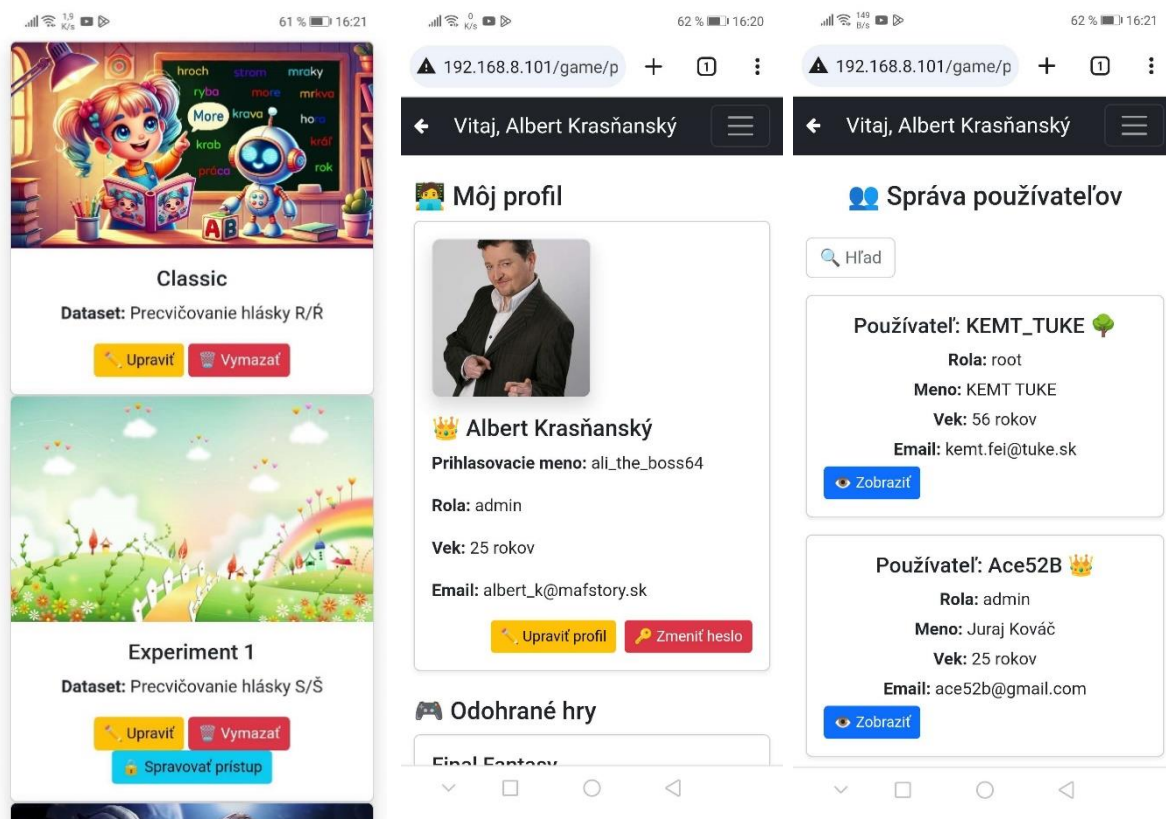


Obr. 105 - Ukážka hry

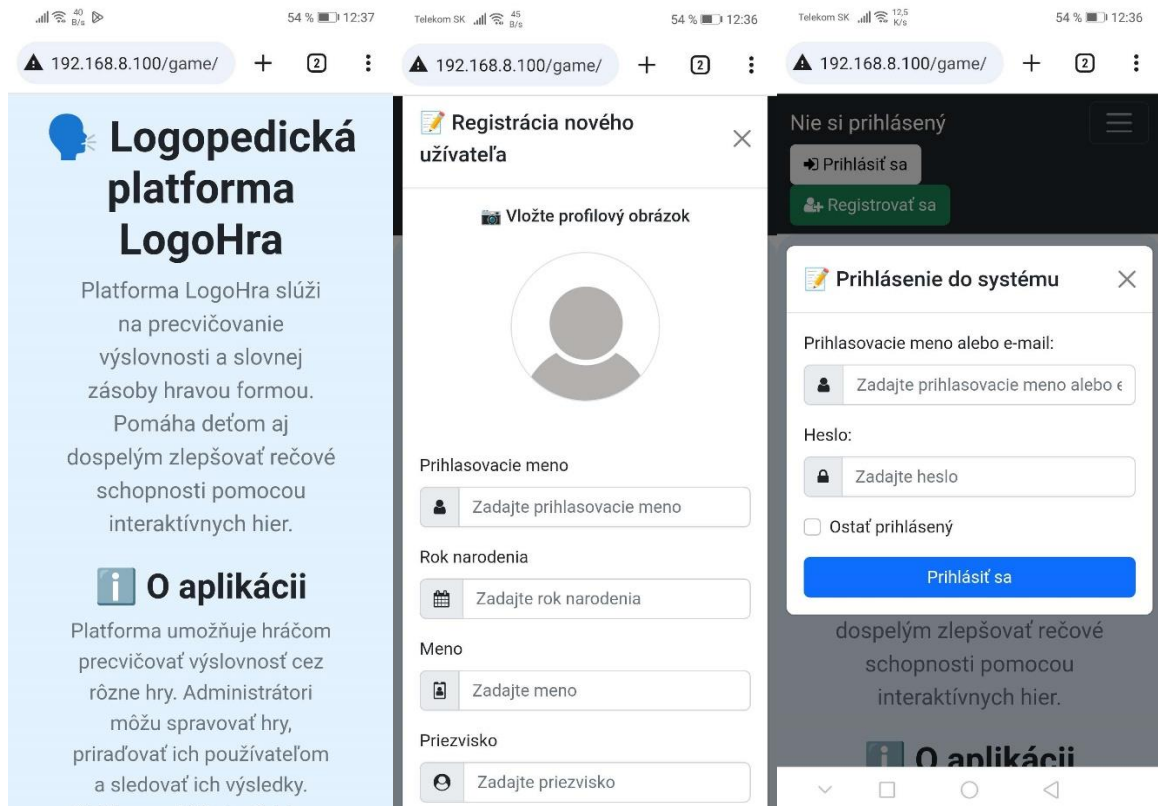


Obr. 106 - Koniec hry

Mobilnú verziu platformy môžeme vidieť na Obr. 107 a Obr. 108. Vidíme, že platforma spĺňa požiadavky na mobilný dizajn, a teda je responzívna pre mobilné zariadenia.



Obr. 107 - Mobilný dizajn



Obr. 108 - Mobilný dizajn pre úvodnú podstránku a formuláre

Záver

Cieľom diplomovej práce bolo bližšie pochopenie gamifikácie a jej potenciálny vplyv na logopedické cvičenia. V teoretickej časti sme si stručne opísali logopédiu a najčastejšie logopedické poruchy, ktoré sa vyskytujú u detí v predškolskom a školskom veku. Následne sme v ďalšej kapitole teoretickej časti zadefinovali pojem hra a počítačová hra, ktoré sú základom porozumenia pojmu gamifikácia. Pre gamifikáciu sme zadefinovali históriu, pravidlá a aké sú jej výhody a nevýhody. Na záver sme ukázali aktuálne využitie v mobilných aplikáciách, ale aj v terapeutických cvičeniach a potenciálne využitie v budúcnosti.

Po naštudovaní pojmu gamifikácia sme začali s tvorbou webovej platformy LogoHra, ktorá slúži na gamifikovanie logopedických cvičení hravou formou. Stanovili sme si, že hlavnou vlastnosťou platformy je dynamickosť a variabilita, čím sme docielili, že do platformy je možné vytvoriť hru, ktorá môže byť rozdielna od iných hier. Tvorbu sme začali popisom technológií ako HTML, CSS, JavaScript, PHP a pod. Následne sme vytvorili návrhy a dizajny pre každú podstránku platformy. Dokopy sme vytvorili 11 dizajnových návrhov, ktoré poslúžili ako pomôcka pri skutočnej tvorbe.

Po realizácii dizajnu sme začali s prácou na samotnej platforme. Ako prvú časť sme realizovali administrátorskú časť aplikácie, ktorej úlohou bola správa jednotlivých úloh (pridanie, úprava, odstránenie), z ktorých následne sa môže vytvoriť dataset a samotná hra. Druhou časťou bolo vytvorenie logiky samotnej hry. Hra pozostávala z herného poľa, po ktorom sa hráč alebo hráči pohybujú. Každé políčko patrilo do skupiny rozdelených podľa farieb. Každá farba reprezentovala iný typ úloh. Úlohy boli prispôbené tak, aby boli jednoduché pre deti a aby boli čo najviac rozpoznateľné. Na úlohy sa následne odpovedalo hlasovým vstupom, ktorý bol konvertovaný do textovej formy pomocou Web Speech API a následne porovnaný so správnou hodnotou uloženou v datasete.

Na záver sme vykonali vytvorenie demo hry Classic, ktorá poslúžila ako demonštrácia platformy. Po vytvorení demo sme túto platformu nasadili na školskú sieť TUKE, kde prebehol test dizajnu a funkcionality na vzorke piatich ľudí, ktorý dobrovoľne vyskúšali túto platformu. Platforma prešla testom, kde sa našli menšie nedostatky v dizajne. Avšak platforma má stále svoje nedostatky ako nesprávna podpora hry na mobilných zariadeniach. V rámci dizertačnej práce by sme chceli využiť súčasný stav platformy ako základ pre jej rozšírenie. Rozšírenie by mohlo spočívať v nových režimoch, či už pre aktuálny typ hry alebo vytvorením nového režimu, ktorý bude viac dynamickejší, zábavnejší a dizajnovovo priateľský. Administrátorská časť by sa mohla rozšíriť o grafické zobrazovanie výsledkov pomocou grafov alebo potenciálnym nasadením umelej inteligencie pre analýzu výsledkov. Na záver plánujeme nasadenie webovej platformy do reálnej internetovej prevádzky.

Zoznam použitej literatúry

- [1]. Kováč, Juraj. (2023). Podporný nástroj pre detskú logopédiu. [Bakalárska práca]. Technická univerzita v Košiciach, Fakulta elektroniky a informatiky. [Citované dňa: 15.3.2025].
- [2]. Šimčíková, Lea. (2022). Technologická podpora logopedickej terapie. [Diplomová práca]. Technická univerzita v Košiciach, Fakulta elektroniky a informatiky. [Citované dňa: 15.3.2025].
- [3]. Oravcová, Mária. (2019). Meranie rečových komunikačných schopností otologických pacientov. [Diplomová práca]. Univerzita Pavla Jozefa Šafárika v Košiciach, Filozofická fakulta.
- [4]. Hirsch, Hans-Günter, Neumann, Christian, Tiggelkamp, Yannic, Fiorista, Riccardo, Knecht, Stefan, Schnitzler, Alfons, Biermann-Ruben, Katja, Bothe, Dietmar, Bleimann, Günter, Frieg, Hendrike. (2023). RehaLingo – Towards a Speech Training System for Aphasia. 34. Konferenz Elektronische Sprachsignalverarbeitung. s. 134-141. Dostupné z: https://www.essv.de/pdf/2023_134_141.pdf. [Citované dňa 15.3.2025].
- [5]. Stoffová, Veronika. (2016). Počítačové hry a ich klasifikácia. s. 243-251. [Online]. Dostupné z: <https://tvv-journal.upol.cz/pdfs/tvv/2016/01/36.pdf>. [Citované dňa: 15.3.2025].
- [6]. MasterClass, Jún 7, 2021. Guide to Video Game Genres: 10 Popular Video Game Types. [Online]. Dostupné z: <https://www.masterclass.com/articles/guide-to-video-game-genres>. [Citované dňa: 15.3.2025].
- [7]. Matsubara, Kenji. Jún 24, 2024. The History of Computer Games and the Birth & Future of Esports. [Online]. Dostupné z: <https://www.lesrencontreseconomiques.fr/2024/en/debats-idees/the-history-of-computer-games-and-the-birth-future-of-esports/>. [Citované dňa 15.3.2025].
- [8]. Arenas, Daniel Lucas, Viduani, Anna, Araujo, Renata Brasil. (2022). Therapeutic Use of Role-Playing Game (RPG) in Mental Health: A Scoping Review. Simulation & Gaming, 53(3), s. 285-311. Dostupné z: <https://doi.org/10.1177/10468781211073720>. [Citované dňa: 15.3.2025].
- [9]. Sorkio, Joonas. (2021). Immersive experiences in racing video games. [Bakalárska práca]. Tampere University, Faculty of Information Technology and Communication Sciences. Dostupné z: <https://trepo.tuni.fi/bitstream/handle/10024/136286/SorkioJoonas.pdf?sequence=2&isAllowed=y>. [Citované dňa: 15.3.2025].
- [10]. Buro, Michael. (2003). Real-Time Strategy Games: A New AI Research Challenge. [Online]. Dostupné z: <https://skatgame.net/mburo/ps/RTS-ijcai03.pdf>. [Citované 15.3.2025].

- [11]. Dormans, Joris. (2010). Adventures in level design: generating missions and spaces for action adventure games. [Online]. Dostupné z: <https://doi.org/10.1145/1814256.1814257>. [Citované dňa: 15.3.2025].
- [12]. Mezeiová, Adriana. (2017). Gamifikácia ako nástroj vzdelávania 21. storočia. s. 270-279. [Online]. Dostupné z: <https://relik.vse.cz/2018/download/pdf/192-Mezeiova-Adriana-paper.pdf>. [Citované dňa: 16.3.2025].
- [13]. Cintulová Lucia Ludvig. (2022). Gamifikácia ako nástroj k zvyšovaniu kvality neformálnej environmentálnej výchovy, vzdelávania a osvetu. s. 8-52. [Online]. Dostupné z: https://old.ewobox.sk/uploads/source/tmp_2022_11_14_09_25_07_69730a164a5006f77d73a85575846f54.pdf. [Citované 16.3.2025].
- [14]. Urbanová Elena. (2018). Čo je to gamifikácia a oplatí sa ju použiť?. [Online]. Dostupné z: <https://www.e-learnmedia.sk/blog/co-je-to-gamifikacia-a-oplati-sa-ju-pouzit/>. [Citované dňa: 16.3.2025].
- [15]. MAXIMUS. (2020) Význam gamifikácie vo vzdelávaní a zapojení študentov. [Online] Dostupné z: <https://maximusproject.eu/sk/vyznam-gamifikacie-vo-vzdelavani-a-zapojeni-studentov/>. [Citované dňa: 16.3.2025].
- [16]. Sloneek. (2022). Čo je to gamifikácia?. [Online] Dostupné z: <https://www.sloneek.sk/slovník-po-jmu/gamifikacia/>. [Citované dňa: 16.3.2025].
- [17]. Copymate. (2024). Gamifikácia v biznise – využitie herných mechanizmov na zapojenie používateľov a zamestnancov. [Online]. Dostupné z: <https://copymate.app/sk/blog/multi/gamifikacia-v-biznise-vyuzitie-hernych-mechanizmov-na-zapojenie-pouzivatelov-a-zamestnancov/>. [Citované dňa: 16.3.2025].
- [18]. Marques Gonçalves, Bhoi Akash Kumar, Albuquerque Victor Hugo C. de. (2021). IoT in Healthcare and Ambient Assisted Living. Studies in Computational Intelligence. s. 223-241 [Online]. Dostupné z: <https://doi.org/10.1007/978-981-15-9897-5>. [Citované dňa 16.3.2025].
- [19]. Fleming, Theresa M., Bavin, Lynda, Stasiak, Karolina, Hermansson-Webb, Eve, Merry, Sally N., Cheek, Colleen, Hetrick, Sarah. (2017). Serious Games and Gamification for Mental Health: Current Status and Promising Directions. Frontiers in Psychiatry. [Online]. Dostupné z: <https://doi.org/10.3389/fpsy.2016.00215>. [Citované dňa: 16.3.2025].
- [20]. Janssen, Joep, Verschuren, Olaf, Renger, Willem Jan, Ermers, Jose, Ketelaar, Marjolijn, van Ee, Raymond (2017). Gamification in Physical Therapy. Pediatric Physical Therapy, s. 95-99. [Online]. Dostupné z: <https://doi.org/10.1097/pep.0000000000000326>. [Citované dňa: 16.3.2025].
- [21]. Afyouni, Imad, Rehman, Faizan Ur, Qamar, Ahmad Muaz, Ghani, Sohaib, Hussain, Syed Osama, Sadiq, Bilal, Basalamah, Saleh (2017). A therapy-driven gamification framework for

- hand rehabilitation. *User Modeling and User-Adapted Interaction*, s. 215-265. [Online]. Dostupné z: <https://doi:10.1007/s11257-017-9191-4>. [Citované dňa: 16.3.2025].
- [22]. Thakur, Kanishka. (2024). Future Of Gamification 2024 Trends. [Online]. Dostupné z: <https://www.nudgenow.com/blogs/future-of-gamification-trends>. [Citované dňa 16.3.2025].
- [23]. Obrázok zo zdroja: <https://fr.vecteezy.com/art-vectoriel/41665125-3d-isometrique-plat-vecteur-conceptuel-illustration-de-ludification-interactif-contenu>. [Citované dňa 16.3.2025].

Prílohy

Príloha A: Webová aplikácia – Príloha obsahuje kompletnú webovú aplikáciu so všetkými súbormi (HTML, CSS, JS, PHP, MySQL skripty).

Príloha B: Obrázky pre aplikáciu – Príloha obsahuje 127 obrázkov vo formáte .png + Zoznam využitých a nevyužitých obrázkov pre aplikáciu vo formáte .xlsx.

Príloha C: Používateľská príručka – Obsahom používateľskej príručky je návod pre správne vytvorenie hier vo webovej aplikácii.