

UNIVERZITA KOMENSKÉHO V BRATISLAVE

FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

WEBOVÁ APLIKÁCIA NA PODPORU  
VZDELÁVANIA ŽIAKOV V PRVOM ROČNÍKU  
ZÁKLADNÝCH ŠKÔL

Bakalárska práca

2012

Paula Schichmanová

UNIVERZITA KOMENSKÉHO V BRATISLAVE

FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

WEBOVÁ APLIKÁCIA NA PODPORU  
VZDELÁVANIA ŽIAKOV V PRVOM ROČNÍKU  
ZÁKLADNÝCH ŠKÔL

Bakalárska práca

Študijný program: Informatika  
Študijný odbor: 2508 Informatika  
Školiace pracovisko: Katedra Informatiky  
Školiteľ: RNDr. Richard Ostertág PhD.

2012

Paula Schichmanová



Univerzita Komenského v Bratislave  
Fakulta matematiky, fyziky a informatiky

---

## ZADANIE ZÁVEREČNEJ PRÁCE

**Meno a priezvisko študenta:** Paula Schichmanová  
**Študijný program:** informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)  
**Študijný odbor:** 9.2.1. informatika  
**Typ záverečnej práce:** bakalárska  
**Jazyk záverečnej práce:** slovenský

**Názov:** Webová aplikácia pre podporu vzdelávania žiakov v prvom ročníku na ZŠ  
**Cieľ:** Cieľom práce je vyvinúť webovú aplikáciu, ktorá bude slúžiť ako učebná pomôcka pre žiakov prvej triedy základnej školy pri vyučovaní nasledovných predmetov:  
\* Slovenský jazyk, písanie - tréning správneho písania základných ťahov - rozpoznávanie správneho písania  
\* Slovenský jazyk, čítanie - generovanie slov pozostávajúcich zo zvolených písmen a slabík  
\* Matematika - porovnávanie počtu prvkov, sčítanie, odčítanie, zlomky - graficky znázornené s testovaním  
\* Matematika - analógové hodiny, čítanie času  
Program bude vypracovaný s využitím Apache, PHP, MySQL, HTML5 a AJAX.

**Vedúci:** RNDr. Richard Ostertág, PhD.  
**Katedra:** FMFI.KI - Katedra informatiky

**Spôsob prístupnosti elektronickej verzie práce:**  
bez obmedzenia

**Dátum zadania:** 06.10.2011

**Dátum schválenia:** 10.10.2011

doc. RNDr. Daniel Olejár, PhD.  
garant študijného programu

---

študent

---

vedúci práce

## **Pod'akovanie**

Rada by som pod'akovala vedúcemu bakalárskej práce RNDr. Ostertágovi za cenné rady.

Paula Schichmanová

## **Abstrakt**

Cieľom tejto práce je vytvoriť webovú aplikáciu zameranú na žiakov prvého ročníka základných škôl, ktorá má pomôcť s ich výučbou. V práci bude popísané, aké technológie som využila, ako som postupovala pri tvorbe programu a budú tam vysvetlené algoritmy, ktoré som použila.

**Kľúčové slová:** webová aplikácia, základné školy, pomoc pri výučbe

## **Abstract**

Goal of this thesis is to create a web application for first grade elementary school students, to help them with studying. In the thesis will be described the technologies and algorithms I used and also how I approached this topic in general.

**Key words:** web application, elementary schools, help with teaching

# Obsah

Úvod.....	1
1. Technológie .....	3
1.1 jQuery .....	3
1.2 HTML5 .....	4
1.3 PHP.....	5
2. Prvostupňové vzdelávanie .....	7
2.1 ISCED 1.....	7
2.2 Ciele.....	8
2.3 Absolventi.....	8
3. Popis Aplikácie.....	9
3.1 Časť 1 – Hodiny.....	10
3.2 Časť 2 – Matematika .....	11
3.3 Časť 3 – Čítanie .....	12
3.4 Časť 4 – Písanie .....	13
3.5 Databáza .....	13
4. Štruktúra aplikácie .....	15
4.1 Základná štruktúra .....	15
4.2 Index.php .....	16

4.3 Adresár PHP .....	16
4.4 Views .....	17
4.5 Resources .....	19
4.5.1 welcome.js .....	20
4.5.2 clock.js.....	20
4.5.3 numbers.js.....	21
4.5.4 drawing.js.....	23
4.5.5 reading.js.....	25
Záver .....	27
Literatúra.....	28



# Zoznam obrázkov

4.1. Graf znázorňujúci funkciu odmocniny.....	23
4.2. Napojenie čiar s malou hrúbkou.....	23
4.3. Napojenie čiar s väčšou hrúbkou.....	23
4.4. Druhy ukončenia čiar.....	24
4.5. Graf znázorňujúci funkciu $y=3*x^2 - 2*x^3$ na intervale $\langle 0,1 \rangle$ .....	25

# Úvod

V dnešnej dobe je internet súčasťou každodenného života, preto som si zvolila bakalársku prácu, ktorej cieľom je zužitkovať fakt, že internet sa nachádza takmer v každej domácnosti. Výsledkom tejto práce by mala byť webová aplikácia zameraná na žiakov prvého ročníka základných škôl, ktorej cieľom je pomôcť im v ich štúdiu.

Aplikácia bude pozostávať z niekoľkých hlavných častí, každá z nich bude zameraná na získavanie a rozvíjanie zručností žiaka v uvedených oblastiach: čítanie, písanie, matematika a určovanie času na hodinách. Každá z nich, bude pozostávať z dvoch alebo troch úrovní, na ktorých bude môcť užívateľ zlepšovať svoje schopnosti.

Témou tejto práce je vytvoriť internetovú aplikáciu s relatívne jednoduchým interfejsom, aby s ňou boli schopné pracovať bez väčších problémov aj deti. Zároveň by mala v duchu hesla „škola hrou“ pôsobiť motivujúco. Mala by podnietiť snahu zlepšiť sa v daných oblastiach. To sa dá docieľiť napríklad pomocou akejsi formy “odmien” v prípade zlepšenia pri riešení jednotlivých úloh. V internetovej podobe by to mohlo znamenať zvýšenie úrovne dieťaťa v zručnosti v danej oblasti. Druhý spôsob motivácie je súperenie medzi jednotlivými používateľmi. To sa môže docieľiť napríklad pomocou tabuľky najlepších dosiahnutých výsledkov, kde sa budú porovnávať výsledky všetkých používateľov.

Na začiatku tejto práce budú popísané technológie, ktoré budeme využívať: HTML5, jQuery, PHP atď. Keďže ide o aplikáciu pre deti, budú na stránke prevládať farby a obrázky. V takomto prípade je vhodným prostriedkom HTML5 s jeho prvkom `<canvas>` a schopnosťou vykresľovať .svg obrázky. Pri práci s animáciou a AJAXom budeme používať jQuery. PHP budeme využívať hlavne na prístup a prácu s databázou.

Druhá kapitola bakalárskej práce sa venuje požiadavkám Štátneho pedagogického ústavu na absolventov prvého ročníka a celkovo prvého stupňa základnej školy. V tretej kapitole je popísaná aplikácia všeobecne. Opisuje vzhľad aplikácie a venuje sa aj jednotlivým častiam osobitne. Pri každej časti je popísané ako vyzerá, ako funguje a ako sa ovláda. Jednotlivé časti web stránky sa skladajú z dvoch alebo troch úrovní, na ktorých sa dá daná časť hrať. Tieto úrovne sú tiež popísané v tretej kapitole. V závere kapitoly sa nachádzajú informácie o databáze, jej tabuľkách a údajoch, ktoré obsahuje.

Posledná (štvrtá) kapitola sa podrobne venuje štruktúre programu. Podrobne popisuje jednotlivé časti štruktúry a súbory, ktoré obsahujú. Najviac sa venuje JavaScriptovým častiam programu, ktoré tvoria najväčšiu časť programu. V rámci nich práca popisuje jednotlivé funkcie, ktoré obsahujú.

# Kapitola 1

## 1. Technológie

### 1.1 jQuery

jQuery je asi najpoužívanejšia JavaScript knižnica. Je to open-source softvér. Je určený na jednoduchšie vytváranie animácií, AJAX aplikácií a rýchlejšie spracovanie udalostí. Okrem týchto výhod poskytuje aj lepšiu manipuláciu s DOM (Document Object Model – konvencia na reprezentáciu HTML, XML, XHTML objektov), taktiež manipuláciu založenú na CSS selektoroch. Je využívaný vo vyše polovici z 10 000 najnavštevovanejších web stránok. Je tvorený jedným JavaScript súborom v ktorom sa nachádzajú všetky funkcie. Pri tvorbe internetovej aplikácie sa dá táto knižnica načítať z nejakej lokálnej kópie alebo pomocou linky na niektorú z mnohých kópií nachádzajúcich sa na internete.

Vďaka architektúre jQuery je možné k frameworku vytvárať rôzne pluginy, ktorými sa rozšíri jeho funkcionality, najčastejšie ide o pluginy na pomoc s AJAX-om, rýchlejšie spracovanie udalostí alebo správu cookies.

Prvýkrát sa začalo hovoriť o podobnom projekte v auguste 2005. Vtedy vznikli prvé narážky na JavaScriptovú knižnicu, kde by sa používali CSS selektory s oveľa stručnejšou syntaxou, akú používali dovtedajšie knižnice. Ajax bo pridaný 26.2. 2006.

Prvá stabilná verzia vyšla v auguste 2006 a zatiaľ najnovšia vyšla v marci 2012.

## 1.2 HTML5

Cieľom piatej verzie HTML štandardu (HTML1 vzniklo v roku 1990) je zlepšiť podporu multimediálnych aplikácií a zároveň ponechať jednoduchosť kódu. Okrem toho HTML5 je nielen vylepšením pôvodnej verzie (HTML4.1), ale spája sa aj s pôvodnou verziou XHTML. Tým sa vyhýba mnohým syntaktickým chybám vznikajúcim práve pri mixovaní týchto dvoch štandardov. HTML5 začal vznikáť v roku 2004 vo WHATWG (Web Hypertext Application Technology Working Group), zatiaľ čo W3C (World Wide Web Consortium) sa sústredilo na tvorbu štandardu XHTML 2.0. V roku 2009 sa W3C rozhodlo neobnoviť XHTML. V súčasnosti pracujú spolu s WHATWG na HTML5.

HTML5 bol vytvorený aj s ohľadom na to, že bude môcť byť spustiteľný aj na low-powered zariadeniach (ako napríklad smartphony alebo tablety).

Medzi nové funkcie patria multimediálne `<video>`, `<audio>` a `<canvas>` elementy. V tomto prípade budeme najviac využívať element `<canvas>`, keďže ide o aplikáciu s mnohými obrázkami. Nové sú aj elementy `<footer>` (na označenie spodnej časti stránky - päty), alebo `<nav>` (na označenie navigácie), ktoré nahrádzujú často využívané `<div>`.

Medzi nové technológie sa zaradil aj formát SVG (Scalable Vector Graphics). Tento formát tu bol už dávnejšie, ale nikdy nemal dostatočnú podporu na to, aby sa stal štandardom. Väčšina prehliadačov okrem IE ho podporovala, ale od HTML5 štandardu ho podporuje aj ten (IE9). SVG je založené na XML, a viac-menej ho môžeme chápať ako programovací jazyk na grafiku. Na vykresľovanie v SVG stačí vložiť do html kódu elementy:

```
<svg>  
</svg>
```

V tejto bakalárskej práci nebudeme priamo programovať v SVG, ale pomocou vektorového editora Inkscape si vytvoríme obrázky .svg formátu, ktorých kód má po otvorení v textovom editore presne taký tvar ako kód, ktorý by sme museli doplniť do kódu stránky, ak by sme chceli aby sa nám vykreslil daný obrázok. Hlavnou nevýhodou rastrových obrázkov je to, že pri približovaní obrázkov „kockatie“. Oproti tomu je

výhodou používania .svg obrázkov to, že pri približovaní stránky sa nijako neovplyvňuje ich kvalita, keďže prehliadač si ich vykreslí zasa v potrebnej veľkosti.

Ako už bolo v úvode spomenuté, z nových elementov sa v projekte najviac využíva `<canvas>`. Na ten je možné vykresľovať útvary pomocou JavaScriptu, ale aj obrázky ktoré sú uložené na serveri. V tomto prípade je jedno, či sa použijú rastrové obrázky, alebo vektorové .svg obrázky, keďže program si ich vykreslí na canvas tak či tak pixel po pixeli, čím vznikne rastrový obrázok. Jediný rozdiel môže byť v tom, že vektorové obrázky zaberajú na serveri menej miesta.

## 1.3 PHP

PHP je hypertextový procesor, ktorý pred tým, než odošle HTML stránky prehliadaču (klientovi), najprv v nich interpretuje vlastné príkazy.

To, že je jazyk PHP interpretovaný, znamená, že až do okamihu spustenia je uchovávaný v zdrojovom tvare. Nevýhodou oproti kompilovanému kódu je to, že tým, že kód jazyka musí byť najprv preložený interpretom, celé spustenie stránky trvá dlhšie. Kompilovaný kód je kód, ktorý je k dispozícii priamo v spustiteľnom tvare, počítač jeho inštrukciám rozumie a nemusí nič prekladať. Nevýhodou kompilovaného kódu je však to, že tieto programy nie sú prenosné medzi rôznymi operačnými systémami. Ďalšou nevýhodou je, že v prípade zmeny v kóde treba program znovu preložiť, zatiaľ čo v prípade interpretovaných jazykov stačí zmeniť zdrojový súbor.

PHP, tak ako napríklad JavaScript, je teda schopné vkladať časti kódu do hypertextových stránok. Od JavaScriptu sa však líši podstatným detailom – JavaScripty sú interpretované klientom, PHP však serverom. Oba spôsoby majú svoje výhody aj nevýhody.

Výhody pri interpretovaní klientom:

1. Odľahčuje to server. Ak by mal server interpretovať, musel by spustiť interpreter jazyka, a až potom výslednú stránku odoslať klientovi. Takto jednoducho dostane požiadavku a odošle stránku a zvyšok nechá na klienta.
2. Interakcia s objektmi na web stránke. Klient môže reagovať na stlačenie klávesy užívateľom, pohyb myšou atď. Toto je hlavný dôvod, prečo sú klientské technológie vytvárané.

Výhody pri interpretovaní serverom:

1. Jednoduchá interakcia s aplikáciami na serveri. Tým, že netreba prenášať dáta po internete, sa jednoduchšie zabezpečí bezpečnosť.
2. Klientovi sa posiela už len čistá HTML stránka, nie sú náročné požiadavky na prehliadač, môže stačiť aj staršia verzia.
3. Zo servera sa klientovi už neposielajú žiadne skripty, čím sa znižuje aj počet prenesených dát a nároky na klienta (napríklad nemusí podporovať JavaScripty).
4. Užívateľ sa nemôže dostať ku pôvodným zdrojovým kódom, keďže klientovi sa pošle už iba výsledok ich behu. Keď stránku interpretuje klient, všetky zdrojové kódy sú mu priamo poslané.

PHP začalo vznikáť v roku 1994 doma u pána menom Rasmus Lerdorf. Ten si v Perle napísal systém na evidovanie návštevnosti jeho stránky. Keďže používanie Perlu zaťažovalo WWW-server, neskôr prepísal systém do jazyka C. Tento systém sa zapáčil Rasmusovým známym a takto sa postupne rozšíril medzi väčšie množstvo užívateľov. Rasmus preto systém rozšíril, napísal k nemu dokumentáciu a zverejnil pod názvom Personal Home Page Tools (neskôr Personal Home Page Construction Kit). Druhá verzia – PHP/FI 2.0 sa stala ešte populárnejšou. FI znamená Form Interpreter – program ktorý umožnil jednoduchú manipuláciu s údajmi z formulárov a tiež začleňovanie SQL dotazov do stránok. V roku 1998 vznikla verzia 3.0, ktorá bola oveľa rýchlejšia a obsahovala nové funkcie ako napríklad podporu mnohých databázových systémov, objekty alebo cookies. Aktuálna verzia je PHP 5.4.3.

Na PHP existuje veľké množstvo frameworkov. Z najznámejších sú to napríklad Zend Framework alebo framework Symfony.

Pôvodným plánom bolo písať túto prácu pomocou frameworku Symfony, avšak to sa neskôr ukázalo ako zdĺhavé riešenie, aj keď tento framework obsahuje veľa prostriedkov na urýchlenie programovania. Poskytuje napríklad silné prostriedky na prácu s databázou, kde sa programátor aplikácie už takmer vôbec nemusí namáhať, keďže veľa kódu je napísaného za neho. Taktiež aplikácia robená v tomto frameworku má už vopred predpripravenú štruktúru s presne stanovenými miestami, kde sa má aký kód ukladať. Štruktúra je však tak rozsiahla, že to výrazne spomaľuje zobrazovanie aj jednoduchšej stránky. Vzhľadom na požiadavky tejto práce by bolo využitie tohto frameworku pravdepodobne neprimerané.

# Kapitola 2

## 2. Prvostupňové vzdelávanie

### 2.1 ISCED 1

Ako už bolo v úvode spomínané, táto práca je venovaná žiakom prvého stupňa základnej školy, konkrétne žiakom prvého ročníka. Samozrejme, niektoré úlohy môžu trochu prekračovať rámec učiva prvákov, prípadne ešte zasahovať do učiva prípravného ročníka. Cieľom je však vybrať úlohy a ich parametre tak, aby boli dostatočne (ale nie príliš), náročné pre deti okolo veku šiestich rokov. Úlohy sú zamerané na precvičenie, upevnenie a zopakovanie vedomostí, ktoré už žiaci majú.

To, čo sa majú žiaci prvého stupňa základnej školy v priebehu štyroch rokov naučiť, popisuje Štátny vzdelávací program pre prvý stupeň základných škôl (ISCED 1). Ten stanovuje povinné vyučovacie predmety, ktoré sú rozčlenené do jednotlivých vzdelávacích oblastí.

Program primárneho vzdelávania má zabezpečiť hladký prechod žiaka z predškolskej úrovne na školskú. Vzdelávanie má nadväzovať na predchádzajúce skúsenosti žiaka a budovať na jeho osobnom poznaní a vlastných skúsenostiach. Princíp, ktorý program využíva, je postup od známeho k neznámemu. Dieťa nadobúda nové vedomosti z aktivít a udalostí, ktoré sú prepojené s jeho životom, alebo



prostredím, v ktorom vyrastá. Cieľom je zabezpečiť, aby dieťa získavalo vedomosti, zážitky a skúsenosti z hroveho aj učebného prostredia a aby získalo potrebu realizovať sa či už slovne, písomne alebo pomocou obrazov. Počas prvostupňového vzdelávania sa taktiež kladie dôraz na to, aby sa vopred včas určilo, či a aké špeciálne spôsoby vyučovania dieťa potrebuje. Či už ide o včasnú korekciu prípadných znevýhodnení, ale taktiež aj skoré podchytenie nadania, ktoré si v budúcnosti môže vyžadovať špeciálne výchovno-vzdelávacie postupy.

## 2.2 Ciele

Okrem poskytovania vedomostí má primárne vzdelávanie za cieľ aj:

- naučiť deti ako spoznávať vlastné schopnosti
- podporovať žiakov kriticky a tvorivo myslieť
- naučiť ich konať iniciatívne a tiež ich naučiť sebakontrolu a schopnosti dorozumieť sa a porozumieť si
- viesť žiakov k tolerancii iných ľudí a akceptovaniu ich hodnôt
- ukázať im, ako si treba uplatňovať svoje práva a zároveň plniť povinnosti.

## 2.3 Absolventi

Absolvent programu primárneho vzdelania by mal mať osvojené nasledujúce spôsobilosti: okrem osvojenia si používania materinského jazyka sa musí vedieť vyjadrovať súvisle (písomnou aj ústnou formou), vie sa sústrediť, počúvať a tiež adekvátne reagovať. Je schopný vytvárať a udržiavať vzťahy s rovesníkmi, učiteľmi alebo rodičmi. Je schopný aplikovať základné matematické myslenie do situácií z reálneho života. Takisto je schopný používať logické a priestorové myslenie. Je pripravený učiť sa a získané informácie vie správne spracovať a neskôr vhodne využiť.

Vyššie uvedené schopnosti sú len zlomkom toho, čo sa vyžaduje od absolventa prvých štyroch ročníkov základnej školy. Táto webová aplikácia sa sústreďí predovšetkým na žiakov prvého ročníka a bude sa snažiť pomôcť tomu, aby jeho absolvent spĺňal požiadavky dané Štátnym pedagogickým ústavom.

# Kapitola 3

## 3. Popis Aplikácie

Aplikácia sa bude skladať zo štyroch základných častí, z ktorých každá sa bude venovať inej oblasti učiva prvého ročníka základnej školy. V každej časti sa budeme snažiť obsiahnuť učivo tak, aby bolo primerané učivu prvého ročníka a zároveň aby malo viacero úrovní, na ktorých sa dieťa môže cvičiť a tým aj zlepšovať.

V úvode aplikácie sa užívateľ zaregistruje (ak chce, aby si program pamätal jeho úspechy). Po registrácii je na základnej úrovni, takže na začiatku dostáva najľahšie príklady (úlohy). Pri jednotlivých častiach je viacero úrovní, na ktorých môže užívateľ hrať a za to získavať hodnotenie. Prvá úroveň je vo väčšine prípadov nebodovaná a je iba na precvičenie práce s počítačom alebo danej úlohy.

Po zaregistrovaní sa zobrazí úvodná stránka, na ktorej je vypísaná malá tlačaná abeceda. Dieťa má za úlohu označiť písmenká, ktoré už ovláda. Po kliknutí na písmeno sa zmení farba zo sivej na zelenú (alebo naopak).

Aplikácia obsahuje čo najmenej textu, väčšina vecí sa vyjadruje v obrázkoch. Samozrejme, niektoré zadania musia byť popísané slovne. V tomto prípade sa aplikácia spolieha na rodičov, že zadania prečítajú a vysvetlia.

Vzhľad stránky je farebný a hravý a obrázky sú na upútanie pozornosti malých žiakov.

## 3.1 Časť 1 – Hodiny

Táto časť je zameraná na určenie správneho času na hodinkách. Dieťa má na výber medzi troma možnosťami hry.

Prvá je len na precvičenie, prípadne naučenie sa určovať čas. Nachádzajú sa tam hodinky a užívateľ má možnosť pohybovať ručičkami. Na paneli pod hodinkami sa vypisuje čas, aký je práve nastavený na ciferníku hodín.

V prípade točenia minútovou (žltou ručičkou) okolo stredu sa pohybuje aj hodinová (o dvanástinu uhla o ktorý sa pohla minútová). Ak užívateľ pustí minútovú ručičku, čas, ktorý ukazuje sa zaokrúhli na najbližší násobok 5.

Ak sa točí hodinovou ručičkou, minútová sa nehýbe (ak by sa mala hýbať, muselo byt to byť 12x rýchlejšie ako hodinová ručička, čoho výsledkom by mohol byť rušivý efekt). Po pustení hodinovej ručičky zastane tak, aby spolu s minútovou ručičkou ukazovali reálny čas. Napríklad ak je minútová ručička na šestke, hodinová môže zastaviť vždy v strede medzi jednotlivými z 12 možných polôh. Hodinová ručička zastane presne na hodnote označujúcej násobok 5 iba v prípade, že minútová ručička je na dvanástke.

Druhá úroveň vyzerá podobne ako prvá, ale spočíva v tom, že žiak má nastaviť na hodinkách taký čas, aký je zadaný v paneli pod hodinkami. Vzhľadom na to, že pohyby ručičiek na hodinkách fungujú tak, ako je vysvetlené vyššie, najvhodnejší spôsob ako nastaviť čas je najprv nastaviť hodinovú ručičku na hodinu, ktorá je v zadaní a potom minútovú ručičku na správnu polohu podľa minút (hodinová ručička sa potom už nastaví sama podľa minútovej). Časy, ktoré sú generované sú vždy zaokrúhlené na najbližší násobok piatich minút.

Po nastavení času dieťa klikne na tlačidlo „Skontroluj“ nachádzajúce sa vedľa panelu so zadaným časom pod hodinkami. V prípade, ak je čas správny, zobrazí sa v strede obrazovky nápis „Správne“. Po kliknutí naň nápis zmizne a hneď je vygenerovaný ďalší náhodný čas, ktorý treba na hodinkách nastaviť. Poloha ručičiek sa po poslednom ťahu nezmení.

V prípade, ak nastavený čas nie je správny, zobrazí sa nápis „Nesprávne“. Po kliknutí naň nápis zmizne ale zadaný čas sa nezmení a tým nechá dieťaťu možnosť nájsť a opraviť svoju chybu.

Po dokončení desiatich kôl sa zobrazí tabuľka s výsledkami. O tom, ako sa určuje výsledné skóre je napísané viac v kapitole 4. Štruktúra aplikácie.

Tretia úroveň funguje presne naopak ako druhá – Po spustení sú ručičky na hodinkách nastavené na náhodne vygenerovanom čase. Pod hodinkami sa nachádza rámček s dvoma prázdnyimi okienkami, kde dieťa vpíše svoj tip na správne riešenie. Tlačidlo skontroluj funguje tak isto ako v druhej úrovni – v prípade, že dieťa zadá nesprávny čas, poloha ručičiek sa nezmení a je tak možné svoj tip opraviť. V tejto úrovni sa ručičkami na hodinkách nedá hýbať.

Druhá aj tretia úroveň sa hrajú v desiatich kolách. Po úspešnom prejdení desiatich kôl sa v strede obrazovky zobrazí štatistika danej hry. Tam je vypísaný počet chýb, čas, trestné sekundy, výsledný čas a najlepší čas daného užívateľa, ktorý sa získa z databázy.

## 3.2 Časť 2 – Matematika

Časť s počítaním sa delí na dve úrovne.

Prvá z nich je jednoduchšia, ide len o počítanie od jedna do dvanásť a nepočíta sa s číslami, ale s jabĺčkami. Príklady sú na sčítanie a odčítanie. V ľavej časti sú zobrazené jablká v náhodne vygenerovanom počte, avšak nikdy to nie je viac ako 12, takisto ani súčet v prípade, že ide o sčítanie, nie je viac ako 12. Výsledky teda môžu byť v rozpätí od 0 do 12 (nula iba v prípade, že ide o rozdiel dvoch rovnakých čísel).

V pravej časti obrazovky sú tri obrázky a každý z nich obsahuje rôzne množstvo jabĺčok s tým, že na jednom sa nachádza správny počet. Vo zvyšných dvoch nie sú počty úplne náhodné, vždy ide o správny výsledok plus mínus dva, aby riešenie nebolo až také jednoduché (ako napríklad by mohlo byť v prípade, že by išlo o sčítanie dvoch vyšších čísel a vygenerované nesprávne výsledky by boli napríklad jedna a dva).

Druhá úroveň sa delí na tri podúrovne. Ak chce hráč postúpiť do ďalšej, musí vyriešiť správne 10 príkladov za sebou z podúrovne, v ktorej sa práve nachádza. Prvá z nich obsahuje príklady s číslami maximálne po 7, druhá po 10 a tretia po 20. Príklady sú náhodne generované a sú aj na sčítanie aj na odčítanie. Po napísaní výsledku a stlačení klávesy enter sa vypíše nasledujúci príklad. V prípade, že užívateľ odpovedal nesprávne, hra mu vypíše ďalší príklad, ale v nasledujúcich kolách sa určite zopakuje príklad, v ktorom spravil chybu.

Program má v databáze zapísané, na akej úrovni sa hráč nachádza. Príklady generuje tak, aby s vyššou pravdepodobnosťou obsahovali väčšie čísla a častejšie sa tak vytvárali náročnejšie úlohy.

Ak sa chce hráč dostať do vyššej úrovne, musí vyriešiť bez chyby 10 príkladov za sebou. Vtedy sa v databáze inkrementuje hodnota úrovne a hráč rovno pokračuje v počítaní vo vyššej úrovni.

### 3.3 Časť 3 – Čítanie

Keďže v súčasnosti sa prváci učia najprv písať tlačenými písmenami a potom písanými, prvá úroveň má za cieľ naučiť deti oba druhy písma. Prebieha to tak, že na obrazovku sa vykreslí tabuľka s dvoma stĺpcami a 4 riadkami. V prvom stĺpci sú 4 náhodné páry písaných písmen (veľké aj malé písmeno), druhý stĺpec je prázdny. Vedľa tabuľky sú 4 páry tlačených písmen. Úlohou dieťaťa je myšou posunúť tlačené písmená do správnych okienok tabuľky tak, aby sa v riadku nachádzalo rovnaké písmeno napísané písaným aj tlačeným. Táto časť nie je bodovaná a slúži len na precvičovanie.

V druhej úrovni bude program generovať náhodné slabiky (vždy spoluhlásku a samohlásku). Dieťaťu sa stále zobrazí jedna a keď ju prečíta, môže na ňu kliknúť (prípadne stlačiť šípku doprava alebo enter) a vtedy sa vygeneruje nová slabika a vypíše sa na miesto pôvodnej.

Tretia úroveň je zameraná na čítanie celých slov. Podľa toho, aké písmená užívateľ zadal v úvode, sa mu budú generovať slová. Na obrazovke sa vždy zobrazí len jedno a vždy sa skladá len z písmen, ktoré užívateľ ovláda. Po kliknutí na slovo sa slovo zmení. Slová sa vypisujú v náhodnom poradí z databázy. Ak sa na obrazovke zobrazí pomlčka, užívateľ prešiel všetkými slovami z databázy, ktoré obsahujú požadované písmená. Obnovením stránky si môže tieto slová zopakovať, prípadne môže upraviť množinu známych písmen v úvode stránky. Vtedy sa databáza aktualizuje a žiakovi budú generované slová už obsahujúce aj novodoplnené písmená.

V tejto časti aplikácie program nebude schopný skontrolovať správnosť čítania a v tomto sa spoľahne na rodičov, prípadne môže ísť len o precvičovanie dieťaťa, ako rýchlo vie čítať. Po prečítaní slova klikne na šípku ďalej a zobrazí sa nové slovo.

## 3.4 Časť 4 – Písanie

Táto časť bude obsahovať dve úrovne.

Prvá z nich bude len kreslenie na canvas, aby sa dieťa zžilo s tým, ako to funguje. Bude mať na výber viaceré farieb štetca a možnosť zmazania plochy.

Druhá úroveň bude tiež obsahovať canvas, ale bude na ňom už vopred vykreslený vzor, podobný tým, ktoré sú v prváčkových písankách. Aplikácia obsahuje 4 vzory, medzi ktorými si žiak môže prepínať pomocou šípok pod canvasom. Kreslí sa čiernou čiarou rovnakej hrúbky ako je hrúbka predkresleného obrázka. Žiak sa snaží čo najpresnejšie prejsť po danej predlohe. Po pustení myši sa vypíše percentuálna úspešnosť (na koľko percent sa trafil na čiaru). Program tiež vždy vypíše aj percentuálnu úspešnosť používateľovho najúspešnejšieho pokusu, ktorá je pre každú predlohu uložená v databáze. V prípade, že užívateľ svoj rekord vylepšil, prepíše sa údaj v databáze aj na obrazovke.

## 3.5 Databáza

Databáza obsahuje 3 tabuľky – `users`, `data` a `words`. Prvá z nich je tabuľka s údajmi o užívateľovi. Pri každej registrácii sa v nej vytvorí nový riadok a do neho sa uložia údaje zadané užívateľom. V prvom stĺpci je ID, ďalej prezývka, zakódované heslo, osemmiestny salt, krstné meno, priezvisko a škola, ktorú navštevuje.

Počas registrácie pribudne aj riadok v druhej tabuľke `data`. V nej sa budú ukladať úspechy užívateľa v jednotlivých hrách. Vyhľadáva sa v nej pomocou ID v prvom stĺpci, ktoré je rovnaké pre daného užívateľa v oboch tabuľkách.

Názvy stĺpcov v tejto časti majú daný tvar, ktorý sa skladá z názvu časti aplikácie, jej úrovne, typu informácie, ktorú obsahuje a poradového čísla. Napríklad `numbers2_time3` obsahuje čas, za aký užívateľ prešiel tretiu podúroveň druhej úrovne časti počítanie.

V druhom a treťom stĺpci (`clock2_time` a `clock2_tries`) sa nachádzajú výsledky pre druhé kolo v hre s hodinkami. Obsahujú celkový najlepší čas užívateľa (jeho čas plus dve trestné sekundy za každú chybu) a číslo, ktoré určuje, koľkokrát už danú hru hral. To sa po každom prejení desiatich kôl inkrementuje.

Tretí a štvrtý stĺpec s názvami `clock3_time` a `clock3_tries` obsahujú hodnoty ako predchádzajúce dva, ale pre tretiu úroveň hry.

Pri registrácii sa ku časom zadáva hodnota -1 a ku pokusom 0.

Nasledujúcich 5 stĺpcov obsahuje údaje hráča pre hry s počítaním. Prvý z nich má informáciu o najlepšom čase, za akú sa hráčovi podarilo prejsť prvú úroveň (úroveň s jabĺčkami). Ďalší stĺpec obsahuje informáciu pre úroveň 2 a v ňom sa nachádza číslo, ktoré určuje úroveň hráča. Jeho predvolená hodnota je 1.

Stĺpce s názvom `numbers2_time1` až `numbers2_time3` uchovávajú časy v milisekundách, za ktoré užívateľ stihol prejsť jednotlivé podúrovne 1 až 3. V prípade, ak sa užívateľ dostal z podúrovne 1 do podúrovne 2, čas, za ktorý prešiel prvú z nich sa už nezmení, lebo do nižšej úrovne už klesnúť nemôže. Jediný z časov, ktorý sa dá zmeniť je čas tretej podúrovne. Tú môže užívateľ prechádzať ľubovoľne veľakrát. Keďže príklady sú generované stále z rozmedzia čísel  $\langle 0, \text{maximálna\_hodnota} \rangle$ , hráč sa nemusí báť, že si nezopakuje jednoduchšie príklady. Stále je nejaká pravdepodobnosť, že mu ich program vygeneruje.

Nasledujúce 4 stĺpce s poradovými číslami 11. – 14. obsahujú percentuálnu úspešnosť užívateľa pri jednotlivých vzoroch v časti kreslenie. Vždy je tam uložený užívateľov najúspešnejší pokus.

Posledné dva stĺpce s názvami `reading2_cons` a `reading2_vows` obsahujú množiny spoluhlások (consonants) a samohlások (vowels). Každý užívateľ tam má uložené znaky, ktoré ovláda. Tieto množiny sa dajú upraviť na uvítacej stránke jednoduchým zakliknutím jednotlivých písmen. Z týchto stĺpcov čerpá údaje program, ktorý generuje slabiky pre užívateľa na čítanie.

Väčšina údajov z tabuľky `data` slúži hlavne na to, aby užívateľ mohol vedieť, ako sa už zlepšil, prípadne aby mohol porovnať svoje výsledky s ostatnými.

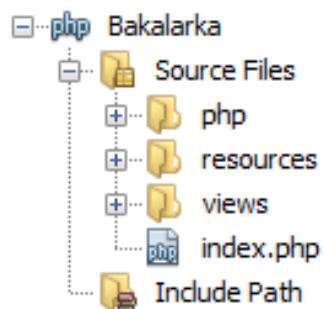
Tretia tabuľka slúži na uchovanie slov v databáze. Tieto slová využíva tretia úroveň časti čítanie. Tabuľka má tri stĺpce. V prvom sa nachádza dané slovo. V druhom je číselná hodnota reprezentujúca samohlásky a v treťom číslo reprezentujúce spoluhlásky.

# Kapitola 4

## 4. Štruktúra aplikácie

### 4.1 Základná štruktúra

Štruktúra kódu aplikácie má takúto podobu:



Všetok kód je rozdelený do troch častí podľa toho, čo vykonáva. Tie popíšem v nasledujúcich podkapitolách.



## 4.2 Index.php

Index.php je jediný súbor mimo adresárovej štruktúry. Je to ten, ktorý sa spustí po otvorení stránky. V ňom sa podľa premennej `$_GET` určí, aký PHP súbor sa má spustiť. Ku každému PHP súboru je priradený JavaScript s rovnakým názvom, aby to uľahčilo volania. Takisto volá aj PHP súbory s názvom `base-top.php` a `base-bottom.php`, kde prvý z nich obsahuje hlavičku a časti stránky, ktoré sa nikdy nemenia. Druhý obsahuje päť stránky. Na jej vytvorenie bol použitý element `<footer>`, ktorý obsahuje meno aktuálne prihláseného užívateľa a tiež aj odkazy na odhlásenie, úvodnú stránku, alebo stránku s najvyššími dosiahnutými výsledkami.

## 4.3 Adresár PHP

V tomto priečinku sa nachádzajú php súbory s funkciami využívanými aplikáciou. Súbor `tools.php` obsahuje základné funkcie na prácu s databázou ako pripojenie do databázy, prihlásenie do aplikácie spolu s kontrolou správnosti údajov zadaných užívateľom. Taktiež obsahuje funkciu na registráciu, v ktorej sa kontroluje, či užívateľ zadal všetky potrebné údaje. V prípade, ak by užívateľ chcel zadať do databázy namiesto mena nejakú časť kódu, je na ochranu použitá funkcia `htmlspecialchars()`.

Na ošetrovanie SQL injections používame `mysqli` (MySQL improved – vylepšenie štandardného modulu na pripájanie) funkciu `real_escape_string()`. `Mysqli` je PHP rozšírenie, ktoré umožňuje prístup k databáze MySQL cez objekty. `Real_escape_string()` je funkcia, ktorá by mala byť používaná vždy pri posielaní SQL požiadavky. Zabraňuje užívateľovi dopĺňať SQL príkazy tým, že pridáva spätnú lomku (backslash) pred vybrané znaky (`x00`, `n`, `r`, `\`, `'`, `“`, `x1a`).

Funkcia, ktorá registruje užívateľa, takisto kontroluje, či už užívateľ s daným menom náhodou neexistuje. Ak sú všetky údaje zadané správne, vygeneruje sa ešte náhodné osemciferné číslo (`salt`) ktoré sa potom pridá na koniec hesla užívateľa a spolu sa na nich použije funkcia `md5()`. Týmto postupom sa výrazne zvýši bezpečnosť hesla. Na internete sa totiž dá nájsť viacero stránok, ktoré obsahujú databázu možných hesiel spolu s ich `md5` zakódovaním. Týmto spôsobom by sa dalo odhaliť mnoho hesiel, ak by sa niekto dostal ku zakódovaným verziám hesiel užívateľov. Po pridaní `salt`-u sa

však zahešované heslo úplne zmení oproti zakódovanému heslu bez salt-u. Každý užívateľ bude mať vlastný náhodne vygenerovaný salt, ktorý sa vždy pri prihlasovaní užívateľa pridá na koniec zadaného hesla a porovná sa s tým v databáze.

V súbore `db_settings.php` sa len priradujú premenným ako názov servera, databázy, heslo a užívateľ príslušné údaje. Posledný súbor je súbor `.htaccess`, ktorý obsahuje len jeden príkaz: `deny from all` (a ten istý sa nachádza aj v priečinku `views`). Tento príkaz zabezpečí to, že užívateľ nemá povolený prístup do daného priečinku.

Súbor `ajax.php` sa vykonáva, ak prišla na server požiadavka asynchrónnej komunikácie (to je komunikácia, počas ktorej má server vykonať nejaké príkazy aj bez načítavania stránky nanovo). Ak prišla takáto požiadavka, v `ajax.php` sa určí, ktorá časť programu danú požiadavku odoslala. Funguje to tak, že v JavaScript kóde, ktorý prináleží jednej časti aplikácie (napríklad `clock.js`), sa pri ajax požiadavkách odošle jeden parameter, ktorý slúži na rozlišovanie (napr. `ajax: "clock2"` znamená, že ide o hru s hodinkami a úroveň dva).

Program si vyčíta z databázy potrebné údaje prihláseného užívateľa. Tie sa porovnávajú s aktuálnymi údajmi, ktoré prišli ako parametre požiadavky na server. V prípade, že aktuálne údaje spĺňajú určité podmienky, zapíšu sa do databázy. Informácie o priebehu spracovania dát sa ukladajú do asociatívneho poľa `$resp`. Na index „ok“ sa uloží či spracovanie prebehlo v poriadku. V prípade, že nastala chyba, je uložená na index „message“. Do poľa sa ukladajú tiež doplňujúce informácie, napríklad, či hráč dosiahol zatiaľ najlepšie skóre alebo koľkokrát už danú hru hral.

## 4.4 Views

Okrem vyššie spomenutého súboru `.htaccess`, `base-top.php` a `base-bottom.php` obsahuje priečinok `views` ďalšie php súbory. Tie sú volané súborom `index.php` podľa toho, akú hodnotu nadobudol atribút `$_GET["page"]`. Tieto stránky sa zobrazujú v aplikácii v hlavnom `DIV` elemente v strede obrazovky. V prípade, že stránka bola nanovo otvorená a atribút `$_GET["page"]` ešte nemá priradenú hodnotu, spustí sa súbor `welcome.php` v ktorom sa nachádzajú formuláre na prihlásenie a registráciu. V opačnom prípade sa spustí jedna zo štyroch stránok

clock.php, drawing.php, reading.php alebo numbers.php, ktoré predstavujú jednotlivé časti aplikácie. Každá z nich ešte obsahuje niekoľko úrovní, na ktorých sa dá hrať. Každá z týchto stránok je vlastne jeden <div>, v ktorom sa nachádzajú jednotlivé elementy, ktoré sú súčasťou hry. V jednotlivých súboroch sa ešte určí, o ktorý level (úroveň) hry ide pomocou atribútu \$\_GET["lvl"]. Každý úrovni je priradená samostatná trieda, vďaka ktorej sa potom dajú úrovne rozlíšiť v JavaScriptových funkciách alebo kaskádových štýloch.

Súbor welcome.php v prípade neprihláseného užívateľa zobrazí prihlasovací a registračný formulár. Po prihlásení užívateľa sa zobrazí privítanie. Program si vytvorí dve polia, jedno na samohlásky a druhé na spoluhlásky. Polia majú namiesto indexov písmená a k nim sú priradené hodnoty false. Potom sa program pripojí do databázy, a z tabuľky data si pre daného užívateľa načíta množinu písmen, ktoré užívateľ pozná. K týmto písmenám zmení hodnotu v poliach z false na true. Potom vypíše jednotlivé písmená do osobitných elementov <span>. Tým, ktoré majú v poli priradenú hodnotu true, dá triedu knows, aby skript v prehliadači vedel, že ide o označené písmená.

Súbor add\_words.php slúži na pridávanie nových prvkov do databázy slov. Najprv získa pole slov z textového súboru slova.txt. Potom si vytvorí pomocné polia samohlások a spoluhlások kde indexy tvoria jednotlivé hlásky a ku každej je priradená hodnota false. Program potom prechádza jednotlivé slová po jednom. Pri každom slove ide písmenko po písmenku a v poli samohlások alebo spoluhlások na daný index priradí hodnotu true. Polia samohlások a spoluhlások sa následne pomocou Hornerovej schémy transformujú do dvoch čísel cyklom:

```
$i = 1;
$vysledok = 1;
foreach ($samohlasky as $value) {
    if ($value)
        $vysledok += $i;
    $i *=2;
}
```

Výsledok je číslo, ktoré v binárnom tvare znázorňuje zastúpenie jednotlivých znakov v slove. Cyklus prebehne dvakrát, osobitne pre samohlásky a osobitne pre spoluhlásky. Vzniknú teda dve čísla, ktoré sa uložia do databázy.

Na príklad uvedieme slovo „bicykel“. Obsahuje samohlásky „i“, „y“ a „e“. Pole samohlások bude teda vyzerat' nejakto: {'a': false, 'e': true, 'i': true, 'o': false, 'u': false, 'y': true}. Po prejdení cyklu nám vyjde ako výsledok číslo 38. To sa v binárnom tvare dá zapísať ako 100110. Z toho vidno, že na polohách reprezentujúcich „e“, „i“ a „y“, sa nachádzajú jednotky, všade inde nuly. To isté platí aj pre spoluhlásky.

Súbor `reading.php` vyčíta z databázy množiny hlások, ktoré prihlásený užívateľ ovláda. Podobne ako program `add_words.php` si vytvorí polia so samohláskami a spoluhláskami na indexoch, kde sa na každom indexe nachádza hodnota false. Potom prejde obe polia a ku písmenám, ktoré užívateľ ovláda priradí hodnotu true. Opäť pomocou Hornerovej schémy vytvorí pomocou týchto dvoch polí čísla, ktoré v binárnom tvare reprezentujú znaky, ktoré užívateľ ovláda.

Číslo, ktoré znázorňuje samohlásky, ktoré užívateľ ovláda, si nazvime `$vnum` a spoluhlásky `$cnum`. Potom už stačí serveru dať požiadavku s podmienkou `WHERE vowels | $vnum = $vnum AND consonants | $cnum = $cnum`. Operátor `|` vykonáva bitovú operáciu OR. Uvedená podmienka vyberie z databázy len tie slová, ktorého písmená sú podmnožinou písmen, ktoré už užívateľ pozná. Napríklad `2 | 5 = 7`, keďže `010 | 101 = 111`. Ak by bol výsledok nejakej operácie OR väčší, ako hodnota reprezentujúca užívateľa, znamenalo by to, že dané slovo obsahuje aj znaky, ktoré užívateľ nepozná. Slová, ktoré spĺňajú požiadavky sa odovzdajú vo forme JSON (JavaScript Object Notation – syntax na ukladanie a výmenu informácií).

## 4.5 Resources

Tento priečinok pozostáva z troch podadresárov.

Prvý z nich s názvom `images` obsahuje všetky obrázky, ktoré sú v aplikácii používané.

Druhý s názvom `stylesheet` obsahuje `.css` súbor v ktorom sú popísané kaskádové štýly pre celý program.

Tretí s názvom `javascript` obsahuje (takmer) všetky JavaScripty použité v aplikácii. Skript `common.js` obsahuje funkcie, ktoré platia pre celú stránku, teda funguje pre tie časti programu, ktoré sa nemenia. Ostatné fungujú tak, že každý z nich prináleží jednému podprogramu venujúcemu sa jednej z častí aplikácie.

### 4.5.1 welcome.js

Obsahuje dve rovnaké funkcie, osobitne pre samohlásky a spoluhlásky. Tieto funkcie fungujú presne naopak, ako PHP funkcia v súbore `welcome.php`. Tá priradila jednotlivým elementom `<span>` triedu `knows` podľa toho, či mala v poli k danému indexu hodnotu `true` alebo `false`. Funkcie v tomto súbore si priradia do polí pre samohlásky a spoluhlásky hodnoty `true` alebo `false` podľa toho, či má element obsahujúci dané písmeno triedu `knows`. Po kliknutí na jeden z elementov sa mu v poli priradí opačná hodnota, ako tam bola pôvodne a zmení sa kaskádový štýl elementu.

Funkcia `$("#send").click()` prejde polia samohlások a spoluhlások, a keď má nejaká hodnota `true`, doplní ju do reťazca, ktorý potom odošle ako jeden z parametrov požiadavky na server.

### 4.5.2 clock.js

Tento skript sa stará o vykresľovanie a ovládanie hier s hodinkami vo všetkých troch úrovniach. Keďže princíp jednotlivých úrovní je viac menej rovnaký, väčšina funkcií funguje pre všetky úrovne. Avšak vo funkcii, kde sa vykonáva pohyb ručičkami, je potrebné kontrolovať, či nie sme v treťom leveli, keďže tam musia byť ručičky fixované. Taktiež treba zabezpečiť, aby sa čas, ktorý náhodne generujeme (tak, aby bol deliteľný 5), zobrazil pri druhej úrovni v podobe čísel pod hodinkami, a v tretej úrovni na hodinkách.

To, na akú ručičku užívateľ klikol, sa zisťuje pomocou súradníc myši pri kliknutí. Tie sa prepočítajú tak, aby sa bod (0,0) nachádzal v strede hodínok. Podľa toho sa vypočíta uhol priamky určenej stredom a bodom kliknutia a porovná sa s uhol jednotlivých ručičiek. V prípade, ak sa ručičky nachádzajú na sebe, porovná sa ešte aj vzdialenosť bodu kliknutia od stredu hodínok. Ak je kratšia ako stanovená dĺžka, kliklo sa na hodinovú ručičku, ak je dlhšia, kliklo sa na minútovú ručičku.

Vykresľovanie prebieha vždy, ak nastane zmena v uhle jednej z ručičiek. Vtedy sa zavolá funkcia `draw()`, ktorej úlohou je vykresliť celé hodinky aj s ručičkami. Vykresľovanie funguje tak, že telo hodínok je uložené ako obrázok vo formáte PNG. Ten sa na canvas vykreslí ako rastrový (aj v prípade, že bol predtým uložený vektorový). Ručičky sa vykresľujú vo funkcii `draw()` ako čiary rôznej hrúbky a dĺžky.

Pri spustení druhej alebo tretej úrovne sa do premennej `start_time` uloží čas, v ktorom bola spustená. Aktuálny čas získame pomocou príkazu `getTime()`, ktorý

vráti počet milisekúnd, ktoré ubehli od 1.1.1970. Po úspešnom prekonaní desiatich kôl sa tento čas odráta od aktuálneho a tým sa získa počet milisekúnd, koľko trvalo užívateľovi na ukončenie úrovne.

Užívateľ vždy po zadaní jeho tipu na výsledok stlačí tlačidlo „Skontroluj“. Vtedy sa zavolá funkcia, ktorá overí, či je jeho výsledok správny a tiež či už nepadol správny výsledok 10 krát. Ak padol, odošlú sa na server informácie o tom, ako dlho to užívateľovi trvalo, o ktorú úroveň ide a počet chýb, ktoré urobil. Do databázy sa ku jednotlivým úrovňam ukladajú 2 veci – aký je užívateľov doteraz najlepší čas, a koľkokrát už danú hru hral. Celkový čas užívateľa sa vypočíta tak, že sa dobe trvania hry prirátajú 2 trestné sekundy za každú chybu, ktorú urobil.

Po prejdení desiatich kôl a stlačení tlačidla „Skontroluj“ sa údaje o hre vypíšu nie len do databázy, ale aj pre užívateľa na obrazovku. V tabuľke bude mať vypísaný počet chýb, jeho čas, ako dlho mu to trvalo, počet trestných sekúnd a výsledný čas, ktorý sa ukladá do databázy. Tieto údaje sa dajú vyrátať pomocou premenných, ktoré používame v programe. Ako posledný údaj v tabuľke sa vypíše užívateľov doteraz najlepší čas. Tento údaj získame z databázy.

Po kliknutí na tabuľku táto tabuľka zmizne, a hra sa začne od začiatku s nulovým časom a žiadnymi chybami.

### 4.5.3 numbers.js

Prvá úroveň matematických úloh je náročnejšia na vykresľovanie ako ostatné, keďže príklady nie sú zadávané pomocou čísel, ale pomocou obrázkov, ktoré sa vykresľujú na canvas.

Funkcia `nacitavanie()` slúži na riešenie problému, ktorý sa vyskytoval zo začiatku pri testovaní stránky. Program niekedy nezobrazil niektorý typ obrázka a vyzeralo to ako úplne náhodný problém, ktorý sa vyskytoval bez akejkoľvek pravidelnosti. Problém bol v tom, že aj keď obrázku bola priradená cesta kde ho hľadať príkazom:

```
obrazok.src = 'resources/images/obrazok.png';
```

v prípade, že si ho stránka nestihla načítať, jednoducho ho nevykreslila a na jeho mieste bol biely obdĺžnik. Riešením je metóda `onload()`, ktorá vykoná nejaký príkaz v prípade, že je obrázok načítaný. Príkaz, ktorý vykoná, je už spomínaná metóda `nacitavanie()`. Táto má zadaný počet obrázkov, ktoré majú byť načítané a vždy

v prípade, ak je zavolaná, inkrementuje premennú, ktorú neskôr porovnáva so správnym počtom. Vykresľovanie sa zavolá až vtedy, ak sa počty zhodujú.

Vykresľovanie prebieha v dvoch častiach na jeden canvas. V prvej časti sa vykreslí ľavá časť rovnice. Obrázky sa vykresľujú do dvoch stĺpcov (maximálny počet v jednom z nich je 6) za nimi nasleduje znamienko, potom opäť obrázky v dvoch stĺpcoch a znamienko rovná sa. Druhá časť vykresľuje jednotlivé možnosti výsledkov.

Program má v poli uložené jednotlivé odpovede a v osobitnej premennej má uloženú polohu správnej odpovede. Po kliknutí na canvas si overí, či bolo kliknuté na jednu z odpovedí a ak áno, tak či na tú správnu. Overovanie prebieha tak (keďže každá možnosť zaberá tretinu výšky canvasu), že stačí vydeliť y-ovú súradnicu kliknutia tretinou výšky canvasu. X-ovú súradnicu majú všetky možnosti rovnakú, preto ju len stačí porovnať s vopred danou konštantou. Element `<div>`, ktorý užívateľovi vypíše, či bola jeho odpoveď správna alebo nie, sa zobrazí v strede obrazovky. Po kliknutí naň zmizne a zadanie príkladu sa zmení, a to aj v prípade, že užívateľ neodpovedal správne.

V druhej úrovni si program najprv zistí, na akej podúrovni sa užívateľ nachádza. Podľa toho priradí do premennej `rozmedzie` maximálnu hodnotu generovaných čísel (prvá úroveň – 7, druhá úroveň – 10, tretia úroveň – 20). Funkcia `vytvorPríklady()` potom vygeneruje, aké znamienko bude v príklade. Ak je to mínus, vygeneruje sa najprv menšenec. Potom sa generuje menšiteľ a výsledok vznikne odčítaním. Pri sčítaní sa najprv vygeneruje výsledok, potom jeden sčítanec tak, aby bol menší od výsledku a druhý sčítanec vznikne ich odčítaním.

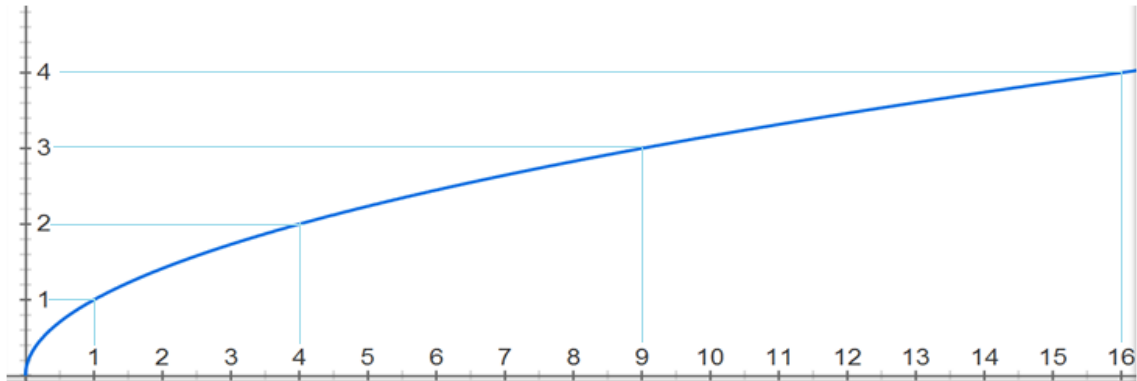
Pri generovaní menšenca (pokiaľ ide o odčítanie) a výsledku (sčítanie) mohla byť vygenerovaná s rovnakou pravdepodobnosťou akákoľvek hodnota z daného rozsahu. No v prípade, že by to bola hodnota 1, máme značne obmedzené možnosti, ako takýto výsledok vznikne (0+1, 1+0). Pri rastúcom výsledku rastie aj počet možností. Potrebovali sme teda generátor čísel, ktorý s väčšou pravdepodobnosťou vygeneruje vyššiu hodnotu. To sa nám podarí pomocou príkazu:

```
vysl = Math.floor(Math.sqrt(Math.random()*(rozmedzie*rozmedzie - 0.01)))+1;
```

Funkcia `Math.random()` generuje náhodné číslo v intervale  $<0,1>$ . Vynásobíme ho druhou mocninou maximálnej hodnoty ktorú chceme získať, takže získame číslo z interval  $<0, rozmedzie^2 - 0,01>$ . Hodnotu 0,01 odpočítame preto, aby sme dostali interval čísel približne  $<0, rozmedzie^2>$ . Tým, že toto číslo odmocníme

a zaokrúhlime nadol, dostaneme s väčšou pravdepodobnosťou vyššie hodnoty ako nižšie.

Pomocou nasledujúceho obrázka sa to dá jednoducho vysvetliť.



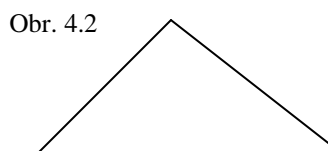
Obr. 4.1 Graf znázorňujúci funkciu odmocniny

Vezmime si napríklad, že chceme vygenerovať výsledok z množiny  $\{1, 2, 3, 4\}$ . Po náhodnom generovaní dostaneme číslo z intervalu  $\langle 0, 15,99 \rangle$ . Po odmocnení získame hodnoty približne  $\langle 0, 4 \rangle$  a po zaokrúhlení nadol to bude číslo z množiny  $\{0,1,2,3\}$ . Ak je to 0, znamená to, že pri náhodnom generovaní a vynásobení sme dostali číslo menšie ako 1. Pravdepodobnosť toho je len  $1/16$ . Ak nám však vyjde výsledok 3, znamená to, že sme vygenerovali číslo v intervale  $\langle 9, 15,99 \rangle$ , čoho pravdepodobnosť je  $7/16$ . Na záver ešte zväčšíme výslednú hodnotu o 1, aby sme dostali požadovanú množinu.

#### 4.5.4 drawing.js

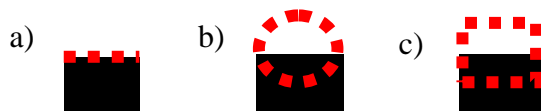
Tak, ako ostatné JavaScript súbory v projekte, aj tento najprv nastaví základné premenné (ako napríklad rozmery canvasu), a potom rozlišuje, ktorá z úrovní je spustená.

Prvá úroveň je iba o kreslení na obrazovku. V prípade kliknutia myšou si program uloží súradnice. Ak sa myšou hýbe, vždy nakreslí čiaru medzi jej najnovšou a poslednou uloženou polohou. Potom zmení hodnoty posledných uložených súradníc na najnovšie a takto sa vykreslí plynulá čiara. Problém nastával pri napájaní čiar väčšej hrúbky na seba. Zatiaľ čo pri hrúbke 1 až 3 pixely problém nebol (obrázok 4.2), pri vyšších hodnotách sa na seba čiary nenapájali plynule (obrázok 4.3):





Tento problém rieši atribút `lineCap`, do ktorého sa môžu priradiť tri rôzne hodnoty – `butt` (obrázok 4.4 a), `round` (obrázok 4.4 b) alebo `square` (obrázok 4.4 c).



Obr. 4.4

V tomto prípade sa nám najviac hodí možnosť `round`, keďže zabezpečí to, aby sa nikde nenachádzali ostré hrany.

Aby kreslenie nebolo pre deti jednotvárne, majú možnosť kliknutím na farebné štvorce zmeniť farbu čiary. To funguje tak, že každý štvorec je v samostatnom `<div>` elemente s priradenou triedou, ktorá má názov ako farba, ktorú daný štvorec reprezentuje. Po kliknutí na štvorec sa farbe čiary priradí názov triedy.

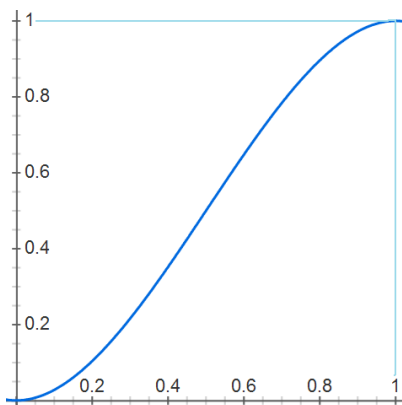
Ak je spustená druhá úroveň, program si načíta obrázok reprezentujúci jeden zo vzorov. Pomocou AJAX-u získa z databázy najlepšie skóre užívateľa na danom obrázku a vypíše ho pod canvas spolu s aktuálnym výsledkom, ktorý je na začiatku 0. Program prejde celý vykreslený obrázok a do premennej si uloží počet sivých pixelov (vzory sú sivé). Po kliknutí a hýbaní myšou sa na canvas vykresľuje čiara. Po pustení myši sa vyhodnotí výsledok. Program opäť prejde všetky pixely obrázku, spočíta všetky sivé pixely a všetky čierne pixely. Výsledok sa vyráta takto:

```
skore = (1-predloha2/predlohaPix)*Math.min(predlohaPix/pero,1) .
```

Premenná `predlohaPix` obsahuje počet sivých pixelov predlohy, `predloha2` obsahuje počet sivých pixelov po prekreslení. Premenná `pero` obsahuje počet čiernych pixelov. Prvý činiteľ predstavuje podiel zafarbených pixelov. Druhý súčiniteľ slúži na to, aby sa program nedal oklamať tým, že by užívateľ zafarbil celú obrazovku. Ak by zafarbil 2 krát viac ako bolo pôvodných sivých pixelov, výsledné skóre sa vynásobí hodnotou 0,5. Ak by zafarbil menej pixelov ako bolo pôvodných, hodnota druhého činiteľa bude 1.

Takéto percentuálne vyjadrenie však môže na deti pôsobiť frustrujúco, keďže sa ťažko dá dosiahnuť skóre nad 80%. Preto sa ešte výsledné skóre pred odoslaním upraví podľa vzorca:

$$\text{vysledok} = 3 * \text{skore}^2 - 2 * \text{skore}^3$$



Obr. 4.5 Graf znázorňujúci funkciu  $y=3*x^2 - 2*x^3$  na intervale  $\langle 0,1 \rangle$

Pomocou tohto vzorca sa vysoké hodnoty viac blížia k maximu a nízke viac k minimu. Aby sa výsledné skóre dieťaťa mohlo čo najviac priblížiť k 100%, vzorec sa na výsledok aplikuje dva krát.

Hneď po tom, keď sa dokreslí čiara, odošle sa požiadavka na server s parametrami: názov hry, výsledné skóre a poradové číslo obrázka. Skript na serveri porovná výsledok s tým, ktorý je uložený v databáze. Ak je nový výsledok lepší, aktualizuje databázu. Pri obnovení stránky alebo prepnutí obrázka sa stále vypíše hodnota, ktorá je aktuálne v databáze.

#### 4.5.5 reading.js

Program si na začiatku uloží do poľa všetky obrázky písaných písmen, ktoré si načíta z priečinka „/resources/images“. Vytvorí si aj druhé rovnako veľké pole `genPismena`, do ktorého na začiatku priradí na každý index hodnotu `false`. Funkcia `generuj()` tomuto poľu priradí na štyri náhodné pozície hodnotu `true`. Táto funkcia tiež vytvorí náhodnú permutáciu štvorprvkového poľa, ktorá bude znázorňovať, ako budú písmenká poprehadzované. Pomocou tejto permutácie sa určia polohy jednotlivých tlačných písmen. Tieto polohy sa uložia do polí `pozicieX` a `pozicieY`.

Potom program odkontroluje, či už boli načítané všetky obrázky. Ak áno, spustí sa funkcia na vykresľovanie. Najprv sa vykresľujú písané písmená. Funkcia prechádza pole `genPismena` a ak sa na nejakom indexe nachádza hodnota `true`, vykreslí obrázok z poľa obrázkov s tým istým indexom. Po vykreslení písaných písmen sa vykreslí mriežka, do ktorej sa písmená budú dopĺňať. Tlačné písmená sa vykresľujú tak, že program opäť prechádza pole `genPismena` a hľadá hodnoty `true`. Keď nejakú nájde, vypíše príslušné písmeno, nie však do radu, ale na polohu určenú poľami `pozicieX` a `pozicieY`.

Ďalšia úroveň si najprv z databázy načíta znaky, ktoré užívateľ ovláda. Výsledkom sú dva reťazce, z ktorých jeden obsahuje samohlásky a druhý spoluhlásky oddelené čiarkou. Tie si rozdelí pomocou funkcie `split(",")` a takto vzniknú polia, kde sa na každom indexe nachádza jedna hláska. Program potom generuje náhodné dvojice hlások, ktoré vždy obsahujú jednu spoluhlásku a jednu samohlásku.

Na stránke sa zobrazujú po jednom. Po stlačení klávesy program skontroluje, či šlo o klávesu enter alebo šípku doprava, ak áno, vygeneruje sa nová slabika.

Dieťa má možnosť aktualizovať množinu písmen, ktoré ovláda kliknutím a slovo „úvod“, ktoré sa nachádza na pätičke dokumentu. To ho odkáže na stránku, kde sa dá zmeniť množina písmen, ktoré má daný užívateľ uložené v databáze.

V rámci tretej úrovne má program uložené pole slov, ktoré spĺňajú podmienky a tým pádom sa môžu zobrazit' užívateľovi. Funkcia `nextWord()` vyberie z poľa náhodné slovo a poznačí si, že už bolo vypísané. Po kliknutí na `<div>` s vypísaným slovom, sa vypíše z poľa ďalšie náhodné slovo. Program si v pomocnej premennej pamätá počet vypísaných slov a ak sa hodnota premennej zhoduje s počtom prvkov v poli, program vie, že už vypísal všetky slová. Vtedy vypíše namiesto slova znak „-“.

# Záver

Internet je každodennou súčasťou našich životov a jeho vplyv na spoločnosť sa neustále zvyšuje. To prináša so sebou nevýhody. Deti už od nízkeho veku pracujú s internetom, ktorý im poskytuje veľa možností na zábavu. Tým strácajú motiváciu rozširovať svoje vedomosti aj vo voľnom čase.

Cieľom tejto práce bolo využiť fakt, že deti školského veku trávajú veľa času na internete a hravým spôsobom ich motivovať k vzdelávaniu.

Výsledkom práce je webová aplikácia pre deti prvého ročníka základnej školy (prípadne predškolákov alebo druhákov). Ovládanie aplikácie sa snaží byť čo najjednoduchšie a práca by mala farbami a vzhľadom upútať detskú pozornosť. Žiaci majú možnosť preskúšať svoje vedomosti zo štyroch oblastí – čítanie, písanie, matematika a určovanie času na hodinkách. Aby to nebolo pre deti jednotvárne, jednotlivé oblasti sú ešte rozdelené do viacerých úrovní podľa náročnosti. Motivovať ich môže tak isto aj fakt, že ich výsledky sa zobrazujú v tabuľke najlepších.

Tým, že sa práca zaoberá vyučovaním, existuje veľa ďalších možností, ako by sa dala do budúcnosti rozšíriť. Jeden spôsob by mohol byť rozšírenie už existujúcich častí aplikácie o nové úrovne. To by mohlo aplikáciu spraviť zaujímavou aj pre starších žiakov. Druhým spôsobom je obohatiť aplikáciu o nové oblasti vzdelávania. Napríklad o učenie angličtiny, prírodovedy alebo vlastivedy (spoznávanie miest na mape, riek, ...).

# Literatúra

- [1] KOSEK, J.: *PHP*. GRADA Publishing, 1999. ISBN 80-7169-373-1
- [2] BRÁZA, J.: *PHP 5*. GRADA Publishing, 2005. ISBN 80-247-1146-X
- [3] <http://trends.builtwith.com/javascript/jQuery>
- [4] <http://en.wikipedia.org/wiki/JQuery>
- [5] <http://www.statpedu.sk/sk/Uvod.alej>
- [6] <http://www.i-programmer.info/programming/graphics-and-imaging/2063-getting-started-with-svg-html5.html>
- [7] <http://en.wikipedia.org/wiki/HTML5>
- [8] <http://www.w3schools.com/js/default.asp>