



Pedagogická  
fakulta  
Faculty  
of Education

Jihočeská univerzita  
v Českých Budějovicích  
University of South Bohemia  
in České Budějovice

Jihočeská univerzita v Českých Budějovicích  
Pedagogická fakulta  
Katedra fyziky

Diplomová práce

# Tvorba elektronických učebních materiálů pro výuku fyziky na ZŠ

Vypracoval: Bc. Vít Barabáš, DiS.  
Vedoucí práce: Ing. Michal Šerý, Ph.D.

České Budějovice 2015

## **Anotace**

Tato práce se zabývá návrhem elektronických učebních materiálů pro výuku fyziky na základní škole. Práce se skládá ze dvou částí. V první části mé práce se věnuji popisu použité technologie HTML5 a jejím možnostem a následně je provedena analýza použitelnosti zvolených technologií.

Hlavní částí práce je návrh a realizace pěti zvolených výukových metod a jejich otestování na běžně používaných interaktivních tabulích a současných mobilních zařízeních.

*Klíčová slova:*

**HTML5, interaktivní tabule, jQuery, Drag and Drop API**

## **Abstract**

The presented diploma thesis deals with designing electronic educational materials intended for learning/teaching physics at elementary schools. The work consists of the following parts. In the first part, we focus on the description of the used HTML5 technology and its possibilities. Subsequently we analyze the applicability of the chosen technologies.

The main part of the work comprises a description of designing and implementation of five selected teaching methods and their testing on interactive whiteboards and contemporary mobile devices.

*Keywords:*

**HTML5, interactive whiteboard, jQuery, Drag and Drop API**

## PROHLÁŠENÍ

Prohlašuji, že svoji diplomovou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. V platném znění souhlasím se zveřejněním své diplomové práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích

dne: .....

.....

Podpis autora práce

## **Poděkování**

Rád bych touto formou poděkoval vedoucímu mé závěrečné práce Ing. Michalu Šerému, Ph.D. za odborné vedení při zpracování mé práce a za cenné rady a doporučení, bez kterých bych tuto práci nenapsal. Dále bych opravdu rád poděkoval panu doc. PaedDr. Jiřímu Tesařovi, Ph.D. za jeho významnou pomoc při návrhu jednotlivých vzorových úloh i za šanci a důvěru do mě vloženou, když si mě vybral pro spolupráci na projektu Interaktivní učebnice fyziky.

Na konec, i když neméně významně bych chtěl poděkovat mým rodičům, kteří mě v průběhu psaní této práce podporovali, a tím mi umožnili práci dokončit. Bez jejich obětavé práce a ustavičné podpory bych nejen že nemohl dokončit svou diplomovou práci, ale ani celá studia.



# Obsah

1	Úvod.....	9
1.1	Cíle práce .....	9
1.2	Metoda vývoje aplikace .....	10
1.2.1	Analýza.....	10
1.2.2	Návrh .....	11
1.2.3	Vývoj.....	12
1.2.4	Implementace .....	12
1.2.5	Evaluace .....	12
2	Terminologie a popis použitých technologií.....	14
2.1	HTML5 .....	14
2.1.1	Specifikace jazyka .....	14
2.1.2	Animace a transformace .....	15
2.1.3	Drag and Drop API.....	16
2.1.4	Podpora v prohlížečích .....	17
2.2	Javascript .....	18
2.2.1	Framework jQuery .....	18
2.2.2	jQuery UI TouchPunch .....	18
2.2.3	Ukládání uživatelských nastavení do COOKIE.....	19
2.3	XML.....	20
2.3.1	Specifikace použití .....	20
2.3.2	Použití v praxi .....	20
2.4	Použité vývojové nástroje.....	21
2.4.1	Eclipse .....	21
2.4.2	Mozilla Firefox .....	22
2.4.3	Verzování CVS.....	22
3	Test technologií na cílových zařízeních.....	24
3.1	Testovací aplikace.....	25
3.2	Test podpory interaktivních tabulí.....	27
3.2.1	Activboard .....	27
3.2.2	Smart Board.....	27

3.2.3	Test podpory na desktopu.....	28
3.2.4	Test podpory na mobilních zařízeních .....	29
4	Vývoj aplikace .....	31
4.1	Návrh úloh .....	31
4.1.1	Myšlenkové mapy .....	33
4.1.2	Pexeso.....	35
4.1.3	DND přiřazení párů .....	36
4.1.4	Křížovka .....	37
4.1.5	Test .....	39
4.2	Návrh jádra aplikace .....	41
4.2.1	Načítání XML.....	41
4.2.2	Generování náhodných čísel .....	43
4.2.3	Mazání prvků z pole .....	43
4.2.4	Kopírování atributů .....	44
4.2.5	Automatické přetypování proměnných .....	44
4.2.6	Vkládání výukových metod.....	45
4.2.7	Načítání výukových úloh.....	45
4.2.8	Výpis dialogů .....	47
4.2.9	Dialog s odkazy na další úlohy.....	48
4.2.10	Zobrazení nápovědy .....	48
4.2.11	Nastavení virtuální klávesnice.....	49
4.2.12	Generování ovládacího panelu .....	49
4.3	Návrh datového modelu úloh.....	49
4.3.1	Základní struktura datových souborů aplikací .....	49
4.3.2	Datový model nápovědy.....	50
4.3.3	Datový model metody křížovka .....	50
4.3.4	Datový model metody DND přiřazení párů .....	51
4.3.5	Datový model metody slepá mapa .....	51
4.3.6	Datový model metody Test .....	52
4.3.7	Datový model metody Pexeso .....	53
4.4	Řetězení úloh .....	54
5	Ověření navržených materiálů při výuce na ZŠ.....	57

6	Závěr .....	58
7	Seznam použité literatury a zdrojů.....	59
8	Přílohy: .....	60



# 1 Úvod

Na počátku psaní této práce jsem byl osloven doc. PaedDr. Jiřím Tesařem, Ph.D., který mi nabídl spolupráci na tvorbě nové interaktivní učebnice fyziky vytvářené pro nakladatelství SPN, a.s.. Pan Jiří Tesař spolu s panem Františkem Jáchymem jsou autory ucelené sady klasických učebnic fyziky pro druhý stupeň základních škol, které jsou zpracované v souladu s RVP. Tyto učebnice se i v současnosti těší velké oblibě mezi učiteli fyziky a jsou používány zhruba na 30 % českých základních škol. Nově vznikající elektronická interaktivní učebnice fyziky bude na tyto učebnice navazovat, ovšem vzhledem k povaze této učebnice, není použití klasické sady učebnic ve výuce pro tuto elektronickou učebnici nutností. Elektronická učebnice může být použita jen jako podpora či zpestření výuky a tak cílí na všechny základní školy v České republice. Tato učebnice bude do jisté míry první opravdu platformě nezávislou a ucelenou elektronickou interaktivní učebnicí v České republice a já se budu v této diplomové práci zabývat vývojem úloh pro tuto elektronickou učebnici.

V mé diplomové práci se zaměřuji převážně na technickou realizaci jednotlivých typů úloh a testování úloh při výuce v praxi. Vzhledem k faktu, že interaktivní učebnice má ambice cílit na všechny základní školy v zemi, značnou částí mé práce je testování vytvořených úloh na zařízeních používaných a dostupných na základních školách a to včetně mobilních zařízení, které budou mít v budoucích letech pravděpodobně čím dál větší uplatnění ve výuce i běžném životě žáků.

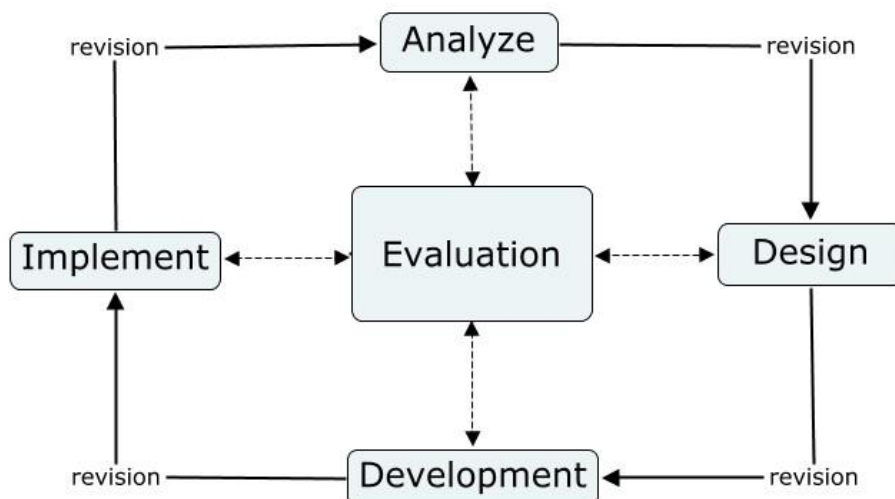
## 1.1 Cíle práce

V mé diplomové práci si kladu za cíl prozkoumat a popsat nástroje pro tvorbu interaktivních elektronických pomůcek s ohledem na jejich použití na PC, tabletu či interaktivních tabulích nejběžněji se vyskytujícími na základních školách v České republice.

Dále bude navržen koncepční model alespoň pěti metod použití vytvořených úloh. Pro navržené metody budou vytvořeny vzorové učební materiály pro výuku fyziky na základní škole. Tyto materiály budou podrobeny testování funkčnosti a použitelnosti na různých platformách. Ověřím použitelnost v operačních systémech MS Windows (7, 8, 8.1), iOS, Android a Linux Ubuntu. Dále ověřím efektivitu navržených materiálů při vlastní výuce.

## 1.2 Metoda vývoje aplikace

Při psaní mé bakalářské práce byla vyvíjena aplikace podle vývojového modelu ADDIE. Při této práci jsem si ověřil výhody tohoto vývojového modelu. Proto bude tento model aplikován i v rámci této diplomové práce. Velkou výhodou je vysoká efektivita vývoje, která těží ze snížení rizika výskytu neočekávaných problémů na naprosté minimum.



OBR. 1 VÝVOJOVÝ MODEL ADDIE ([1])

### 1.2.1 Analýza

Před samotným vývojem jednotlivých výukových metod byla provedena analýza dostupných technologií a jejich možnosti. Při této fázi byla zkompletována literatura, ze které jsem čerpal při psaní této práce. Dále muselo být zjištěno, pro jaká cílová zařízení budou úlohy v učebnici určeny. Vzhledem k tomu, že úlohy zmiňované v této diplomové práci jsou součástí komerčního produktu, musí aplikace splňovat požadavky nakladatelství, pro něž tyto úlohy vznikají.

Základním požadavkem na učebnici je nezávislost úloh na druhu použitého zařízení. Hlavním cílovým zařízením jsou interaktivní tabule. Interaktivní učebnice fyziky bude sloužit učitelům jako rozšiřující nástroj, kterým žákům zpestřuje a zkvalitňuje své hodiny. Na českých základních školách, se můžeme setkat s celou řadou různých druhů interaktivních tabulí od různých výrobců. Bohužel, každý výrobce dodává podpůrný software pro tvorbu úloh optimalizovaný pouze pro své vlastní produkty. Není proto možné, chceme-li vytvořit úlohy funkční a nezávisle na

druhu interaktivní tabule či výrobci dané tabule, využít těchto programů. Jednotlivé druhy tabulí také pracují s dosti odlišnými principy reakcí na podněty uživatele.

Nejběžnějšími interaktivními tabulemi na českých základních školách jsou v současnosti tabule ACTIVboard dodávané firmou Promethean a S.M.A.R.T. Board dodávané firmou AV Media. Interaktivní úlohy budou optimalizované pro provoz právě na těchto dvou zařízeních. Bližšímu popisu se budu věnovat v kapitole *Test podpory interaktivních tabulí*.

Vzhledem k cílům, musela být zvolena technologie, která umožní podporu nejen interaktivních tabulí, ale pokud možno i co možná největšího počtu mobilních zařízení. To sice není požadavkem nakladatelství, ovšem s výhledem na současný vývoj i pohled na znatelný tlak pro zapojení využívání tabletů při výuce, ze strany státních orgánů, to dle mého názoru dává jednoznačný smysl jako příprava na budoucnost. Tento cíl ovšem s sebou přináší i celou řadu potenciálních problémů. Ne nadarmo vznikají vždy aplikace pro konkrétní operační systém, či pro konkrétní zařízení. V současnosti je dostupná pouze jediná technologie, která není závislá na platformě a umožňuje vytvořit aplikaci spustitelnou napříč různými platformami či druhy zařízení a tou jsou internetové stránky. Donedávna by ovšem možnosti internetových stránek neumožnily vytvořit interaktivní aplikace uvnitř stránky. To se řešilo vkládáním různých apletů naprogramovaných v jazyce Java či FLASH. Ty ovšem v drtivé většině případů nefungovaly na produktech firmy Apple a ani ostatní mobilní zařízení jiných značek tento obsah nedokázaly korektně zobrazit. To naštěstí padlo s příchodem jazyka HTML5, který umožňuje autorovi využívat celou řadu efektů a nejrůznějších transformací. Všechny úlohy i samotná aplikace budou realizovány v tomto jazyce (více v kapitole *HTML5*).

V rámci této analýzy jsem vytvořil myšlenkovou mapu, která slouží jako průvodní dokument pro kontrolu plnění dílčích cílů (příloha č. 1).

### **1.2.2 Návrh**

Při této části vývoje, byl v závislosti na provedené analýze, proveden návrh koncepčních modelů pěti metod. Metody byly vybrány po dialogu s autory a vedením nakladatelství SPN.

Vybrány a navrženy byly následující typy úloh:

- myšlenkové mapy
- pexesa
- DND přiřazení párů
- křížovky
- testy

Hlouběji je tato část rozebrána v kapitole 4.2.

### **1.2.3 Vývoj**

Při této části diplomové práce byly naprogramovány jednotlivé navržené úlohy. Úlohy byly programovány co možná nejuniverzálněji, aby šlo jeden kód provozovat v různých prohlížečích. Zároveň byly při této části zjištěny první technické překážky, které bylo potřeba řešit. V této fázi se ukázaly velké rozdíly mezi jednotlivými prohlížeči i systémy a to ať už v technickém řešení, tak i v podpoře, respektive nepodpoře současných uznávaných standardů. Blíže budu popisovat jednotlivé strasti v kapitole pět.

### **1.2.4 Implementace**

Implementace naprogramovaných úloh se ukázala jako jeden z největších problémů. Při vývoji těchto materiálů k dnešnímu dni (1. březen 2015), jsem od autorů dostal zpracovanou necelou jednu kapitolu didaktických materiálů k naplnění věnovanou tématu délka. Při testování navržených úloh na skutečných technických prostředcích, se postupně zjišťovaly další drobné problémy. Nesčetněkrát se z této fáze vývoje produktu projekt vracel do fáze vývoje a byly upravovány již naprogramované úlohy, tak aby splňovaly logická, didaktická a často i jen kosmetická kritéria redaktorů nakladatelství SPN či autorů.

### **1.2.5 Evaluace**

Fáze evaluace nastala po naplnění prvních didaktických materiálů a vytvoření použitelných výukových úloh. Při evaluaci bylo provedeno testování vytvořených materiálů ve dvou fázích. První fází bylo mé osobní testování při příležitosti vykonávání průběžné pedagogické praxe z fyziky, kterou jsem absolvoval na ZŠ Čečova pod vedením pana Mgr. Radovana Mikeše.

Při těchto cenných hodinách bylo mnoho šancí si ověřit použitelnost výukových materiálů v praxi při reálných výukových situacích. Díky této praxi byly zjišťovány spíše technické problémy, které byly průběžně opravovány. Druhým neméně podstatným aspektem zjištěným při této práci, byla práce s barvami. Při této práci byl jasně zřejmý rozdíl mezi různými technickými prostředky. Na ZŠ Čecova byla učebna vybavena interaktivní tabulí ACTIVboard. Tato tabule dosahuje opravdu úctyhodných viditelností a obraz podávaný projektořem byl vždy ostrý a barvy věřné.

Naopak při své souvislé pedagogické praxi, kterou jsem absolvoval na základní škole v Nových Hradech, jsem si vyzkoušel využití vytvořených materiálů na opravdu slabém a již značně vysvíceném projektořu, který interaktivitu zajišťoval pouze prostřednictvím polohovacího pera nahrazující počítačovou myš. Úlohy, které byly bez problému použitelné na ACTIVboardu byly na mnou zmiňovaném projektořu nečitelné už z prostředních řad.

Poslední částí testování, které bylo provedeno, bylo testování materiálů na fakultní interaktivní tabuli značky S.M.A.R.T Board. Na této interaktivní tabuli jsme testovali vytvořené aplikace spolu s panem doc. Tesařem. Protože didaktický obsah je plánovaný pro první polovinu školního roku v šestém ročníku, byl problém najít zrovna tyto třídy. Tento problém byl nakonec vyřešen vytvořením didaktického obsahu vhodného pro mou výuku v rámci jednotlivých praxí.

Druhou fází testování mnou vytvořených materiálů bylo poskytnutí těchto materiálů doktorandům z naší katedry pánům Mgr. Vladimířu Vochozkovi a Mgr. Vítu Bednářovi. Požádal jsem tyto učitele o provedení pedagogické sondy použitelnosti mnou navržených materiálů při jejich výuce na ZŠ. Blíže se této části věnuji v kapitole pět, kde jsou rozebrány zpětné vazby, které jsem dostal.

## 2 Terminologie a popis použitých technologií

Tato kapitola se věnuje základní terminologii použité v této práci a popisu použitých technologií. Jednotlivým částem bylo věnováno již nespočet jiných prací, a proto se omezím pouze na vlastní pohled či důvody použití jednotlivých technologií či technických prostředků použitých při tvorbě této práce.

### 2.1 HTML5

HTML 5 je aktuálně nejnovější standardizovanou technologií pro tvorbu internetových stránek, respektive umožňující tvorbu multiplatformních aplikací. Základem tohoto značkovacího jazyka je sada elementů a atributů standardizovaných konsorciem *w3.org* plus sadou podpůrných technologií jako jsou např. CSS3, XMLHttpRequest2 a nebo nové API pro jazyk Java Script.

#### 2.1.1 Specifikace jazyka

Tento jazyk byl jeho tvůrci standardizován a označen jako dokončený dne 23. dubna 2014 [2] a zároveň byl stanoven výhledový plán pro pokračování návrhu jazyka HTML 5.1. Tento plán potvrdil dokončení návrhu standardizace jazyka, což ovšem, musím dodat bohužel, nezaručuje, že vše navržené v tomto standardu bude ihned podporované napříč běžně se vyskytujícími prohlížeči. V tomto shledávám asi největší nevýhodu této technologie jako celku. Z hlediska časové náročnosti je minimálně 1/3 času práce programátora zabraná optimalizací kódu pro jednotlivé prohlížeče a to jen tím, aby dané funkce dělaly všude to, co mají dělat a aby to dělaly pokud možno stejně. Jazyk HTML5 je stejně jako ostatní HTML jazyky jazykem interpretovaným, a tak zdědil od svých předchůdců celou řadu výhod i problémů. Tím, že je jazyk interpretován prohlížečem, nemůže programátor zaručit nikdy na 100 %, že se obsah jím naprogramovaný zobrazí přesně, tak jak zamýšlel.

#### Výhody použití jazyka

- jeden kód pro všechna koncová zařízení
- výpočet aplikace probíhá na straně klienta
- platformně nezávislý jazyk

## Nevýhody nasazení jazyka

- kód aplikace je veřejně dostupný (tento bod je opravdu hodně bolestivý hlavně u komerčního produktu)
- nelze zaručit s jistotou, že se bude obsah vždy zobrazovat korektně a to i vzhledem do budoucnosti
- interpretovaný jazyk je vcelku pomalý a u rozsáhlejších projektů to může znamenat sekání aplikací na pomalejších strojích
- hrozí riziko nefunkčnosti obsahu, pokud uživatel bude mít nastavený zákaz vykonávání scriptů

### 2.1.2 Animace a transformace

Jazyk HTML5 nově umožňuje použití celé řady pokročilých efektů. V této práci jsou použity nejvíce 2D animace pro zvětšení objektů, přesouvání objektů a nebo 3D rotace. Protože při práci je využíváno javascriptového frameworku jQuery, tak je animování opravdu jednoduché. Nejlépe poslouží příklad z myšlenkových map. Pro jednotlivé aktivní prvky, které lze vyplnit nastavím metody animate a jejím parametrem je objekt<sup>1</sup>, který definuje jak má animace vypadat. S tímto nástrojem je jakákoliv další animace otázkou nastavení jednotlivých parametrů. Konkrétní kód pro aktivaci aktivního prvku vypadá takto:

```
bublina.animate({
  top: bublina.attr('data-start_pos_top'),
  left: bublina.attr('data-start_pos_left'),
  rotate: bublina.attr('data-start_rot') + 'deg',
  scale: bublina.attr('data-start_scale'),
  opacity: '1'}, 1000, function() {
  // uvolnit pro kliknutí
  bublina.attr('data-clicked', 'false');
  lock = false;
  bublina.css('z-index', '100');
});
```

---

1 Přehled všech možných nastavení animací je dostupný z adresy: <http://api.jquery.com/animate/>

Jako ukázka 3D animace by mohlo sloužit přetáčení karet pexesa při reakci na kliknutí.

Kód pak vypadá takto:

```
$(t).find('.rub_karta').css({
    transformOrigin: '0px ' + width / 2 + 'px'
}).animate({
    perspective: '10000px',
    rotateX: '180deg',
    opacity: 0
}, function(){
// karta skovaná
    $(this).hide();
});
```

Kombinacemi nastavení jednotlivých parametrů, nastavovaného objektu parametrů animace, lze vytvořit opravdu bohatou paletu efektů.

Na druhou stranu by se s efekty nemělo plýtvat. Je velice tenká hranice mezi zajímavým obsahem a rušivým přeplácaným neustále se někde pohybujícím obsahem. Pro výukový materiál to pak platí dvojnásob. Tato funkcionality má velice dobrou podporu napříč prohlížeči a proto ji lze bez problémů použít.

### 2.1.3 Drag and Drop API

Další novou funkcí, kterou může programátor použít, je to, co naprosto běžně využívají nativní desktopové aplikace. Možnost chytit a tažením zajistit spuštění nějakého podprogramu, byla pro HTML dlouhou dobou nemožná. S příchodem jazyka HTML5 existuje několik možností, jak toto v projektu realizovat. Bohužel nativní podpora DND API v jazyce HTML5 ještě není natolik podporovaná, abych tuto cestu mohl využít ([4], str. 32). Problematice podpory nativního DND API v HTML5 byla věnovaná má bakalářská práce, a proto jen popíši mnou použité řešení za pomoci javascriptu s využitím frameworku jQuery. Základem použití je metoda `draggable`. Nastavení je, stejně jako u nastavení animací, realizováno objektem obsahujícím parametry vlastností a událostí<sup>2</sup>.

---

<sup>2</sup> Přehled všech možných nastavení objektu je dostupný z adresy: <http://jqueryui.com/draggable/>



## 2.1.4 Podpora v prohlížečích

V této části budu vycházet ze své bakalářské práce. Podpora jednotlivých komponent je zde poměrně dobře popsána ([4], str. 41).

		Testované funkcionality					
		1)	2)	3)	4)	5)	6)
Testované prohlížeče	IE 8	✗	✗	✗	✗	✗	✗
	IE 9	✗	✗	✗	✗	✓	✓
	IE 10	✓	✓	✓	✓	✓	✓
	FF 3.6	✓	✓	✓	✗	✓	✗
	FF 19	✓	✓	✓	✓	✓	✓
	Chrome 13	✗	✗	✓	✗	✓	✓
	Chrome 25	✓	✓	✓	✓	✓	✓
	Opera 12	✓	✓	✓	✓	✓	✓

Podporováno



Nepodporováno



Testy:
1) táhnutí elementem div v rámci jedné stránky
2) táhnutí elementem div mezi dvěma stránkami
3) přetáhnutí souboru z PC do prohlížeče
4) nahrávání souborů na server za pomoci XMLHttpRequest 2
5) tvorba výřezů za pomoci elementu canvas
6) aplikace efektů za pomoci elementu canvas

Jak je vidět z tabulky, podpora nativního DND API není dobrá. Vzhledem k současné situaci, kdy celá řada základních škol v České republice disponuje starými prohlížeči s Windows XP či Windows Vista a na těchto počítačích není nainstalován žádný jiný internetový prohlížeč než Internet Explorer, tuto technologii nelze použít<sup>3</sup>.

<sup>3</sup> Tabulky jsou součástí mé bakalářské práce a jsou v plném znění dostupné z adresy: <https://wstag.jcu.cz/StagPortletsJSR168/KvalifPraceDownloadServlet?typ=1&adipidno=31140>

## 2.2 Javascript

Javascript je interpretovaný programovací jazyk podporovaný internetovými prohlížeči. V tomto jazyce je naprogramován v práci použitý framework jQuery, ve kterém jsou naprogramované všechny výukové metody.

### 2.2.1 Framework jQuery

Framework jQuery je knihovnou řešící celou řadu potencionálních problémů s nekompatibilitou různých prohlížečů. Tím řeší například problém nepodporovaného DND API v jazyce HTML5 ve starších prohlížečích od firmy Microsoft. Základem celého frameworku je knihovna *jquery.js*<sup>4</sup>, která je dostupná jako Open Source a lze ji tak zdarma použít i pro komerční účely. Pro tento framework je dostupná celá řada dalších rozšiřujících knihoven jako např. jQuery UI<sup>5</sup> a je také volně dostupná. Druhou využívanou knihovnou v mém projektu je knihovna *jquery.mobile.js*<sup>6</sup>. Tato knihovna zajišťuje celou řadu oprav problémů s nekompatibilitou dotykových zařízení, jako jsou tablety či mobilní telefony.

### 2.2.2 jQuery UI TouchPunch

Při testování podpory interaktivních tabulí byl zjištěn jeden velice nepříjemný problém a to, že na tabuli S.M.A.R.T. Board nefunguje Drag and Drop API z frameworku jQuery a to ani po aplikování pluginu jQuery Mobile. Problém je způsoben jinou reprezentací spouštěcí události, kdy senzory interaktivní tabule odesílají do prohlížeče tuto informaci jinou metodou. Poměrně dlouhou dobu jsem s tímto problémem bojoval a nebyl jsem schopen ho vyřešit. Nakonec byl diagnostikován problém odposlechnutí všech přijímaných událostí přicházejících do aplikace z této tabule, ale nedokázal jsem tuto událost přemapovat na korektní událost či zajistit bezpečné rozhodování při změně reakce podle použitého zařízení.

Řešením tohoto problému se ukázala být knihovna jQuery UI TouchPunch<sup>7</sup>, která opravuje a sjednocuje různé nestandardní události a spojuje je v standardní. Autor této knihovny vyřešil elegantním způsobem hned několik problémů s nestandardními událostmi a zároveň zachoval zpětnou kompatibilitu s předchozím kódem. Zjednodušeně lze říci, že přidáním této knihovny do projektu se pouze rozšíří podpora

---

<sup>4</sup> Knihovna je dostupná z adresy: <http://jquery.com/>

<sup>5</sup> Knihovna jQuery UI je dostupná z adresy: <http://jqueryui.com/>

<sup>6</sup> Knihovna jQuery Mobile je dostupná z adresy: <http://jquerymobile.com/>

<sup>7</sup> Knihovna jQuery UI TouchPunch je dostupná z adresy: <http://touchpunch.furf.com/>

mezi těmito dříve nepodporovanými zařízeními. Programátor pak používá pouze standardní události `onClick` a `onTouch`.

### 2.2.3 Ukládání uživatelského nastavení do COOKIE

Vzhledem k faktu, že materiály jsou plánované jako software distribuovaný na CD, které je paměť typu ROM (read only memory / paměť pouze pro čtení), bylo potřeba nějak vyřešit problém s ukládáním nastavení. V současnosti přichází v úvahu pouze jedno řešení a to ukládání nastavení do COOKIE. Ty jsou nástrojem určeným pro uchování malých textových informací po omezenou dobu. Proto je potřeba tento nástroj přenastavit do takové podoby, aby mohl fungovat po neomezenou dobu a data v něm byla serializovatelná.

Pro usnadnění práce se soubory COOKIE je navrženo rozšíření *jquery.cookie.js*. Tento doplněk umožňuje snadnou práci s COOKIE a umožňuje s ní programátorovi pracovat jako s běžnou proměnnou. Od chvíle kdy připojíme tento plugin, můžeme serializovaně pracovat s COOKIE jako běžnou zapisovatelnou paměť přes metodu *COOKIE*. Jak ukazuje následující ukázka, obsahem proměnné může být de facto jakákoli datová struktura.

```
// nastavení cookie  
vk = $.cookie('virtual_keyboard', true);
```

Znak `$` ve frameworku značí základní ukazatel na načtené jádro frameworku. Metoda `cookie` obsahuje dva parametry funkce. Prvním parametrem je řetězec označující název proměnné a je povinný. Druhým parametrem je nepovinná hodnota nspecifikovaného typu a v případě, že je vyplněný, označuje novou hodnotu, která bude nastavena do proměnné. V případě, že druhý parametr není nastaven, metoda `cookie` vrátí jako návratovou hodnotu proměnné hodnotu uloženou v COOKIE. Princip je opravdu jednoduchý. V mém projektu jsou prováděna tato inicializační načtení nastavení najednou při zavádění jádra aplikace. Ve zbytku aplikace se už pracuje s již načtenými nastaveními.

## 2.3 XML

XML neboli Extensible Markup Language je rozšířeným značkovacím jazykem, často používaným pro univerzální datové přenosy. Tento jazyk byl zvolen pro ukládání a organizaci zdrojových dat jednotlivých úloh. Každá konkrétní výuková úloha je tvořena právě jedním datovým souborem ve formátu XML. Existují i úspornější řešení jak načítat různá data jako CVS či JSON. Všechny ovšem postrádají jednu podstatnou věc. Nižší paměťová náročnost je vykoupená sníženou přehledností. Pro mé použití není velikost datových souborů, které jsou mimochodem velké průměrně 2 – 3 kB, žádným problémem. Naopak pokud si bude chtít učitel vytvořit vlastní úlohy, pokud zvládá základy tvorby HTML, nebude mít problém tvořit své vlastní nové úlohy.

### 2.3.1 Specifikace použití

XML je co do použití opravdu velice variabilní jazyk. Na rozdíl od jazyka HTML nesusazuje programátora povinností použití konkrétních elementů či atributů, ale pokud programátor chce, má možnost si definovat i vlastní jazyky skládající se z vlastních elementů a atributů. Programátor může pro takto navržený jazyk vytvořit DTD (Document Type Definition), který za pomoci definice použitelných elementů, umožní provádět automatickou validaci načítaných dat. Těchto definic nebylo v mé práci, využito, ale pokud by se mělo například kontrolovat načítání aplikačních dat z nedůvěryhodných zdrojů, bylo by vhodné načítaná data validovat.

Jak jsem již zmínil základem XML dokumentu je uživatelem definovatelný značkovací jazyk. Z hlediska dat je XML soubor pouhým, sice strukturovaným, ale stále textovým datovým souborem. Každý XML soubor obsahuje právě jeden kořenový element a ten může obsahovat libovolný počet vnořených elementů. Zároveň každý pod element může obsahovat libovolné atributy. Pro potřeby této práce byl navržen vlastní značkovací jazyk.

### 2.3.2 Použití v praxi

V praxi se často používají XML soubory pro přenášení dat mezi různými systémy. Nejlépe jsou možnosti vidět při použití XML jako datových mostů. Různé porovnávače cen jako Heureka.cz na svém webu definují formát dat, který přijímají a libovolný výrobce informačního systému, nezávisle na používané technologii, může vygenerovat ze své databáze datový soubor ve formátu XML. Porovnávač si potom tento soubor pravidelně stahuje a ačkoliv porovnávač nemá přímý přístup k danému

systemu, je schopen získat požadované informace a zpracovat je do své databáze. Další výhodou formátu XML je dostupnost celé řady již hotových parserů, díky kterým programátor může k datům přistupovat zcela unifikovaně. Úplně tak odpadá nutnost řešit samotné načítání datových souborů. Pro přístup k takto strukturovanému datovému souboru pak slouží speciálně navržený dotazovací jazyk XQuery a XPath. V jednoduchosti lze připodobnit tento jazyk například k linuxové kořenové struktuře, kdy cesta k hledaným datům je reprezentovaná jako seznam za sebou jdoucích úrovní názvů jednotlivých pod elementů, jen místo znaku / je použita mezera:

```
$(data).find('data variables variable');
```

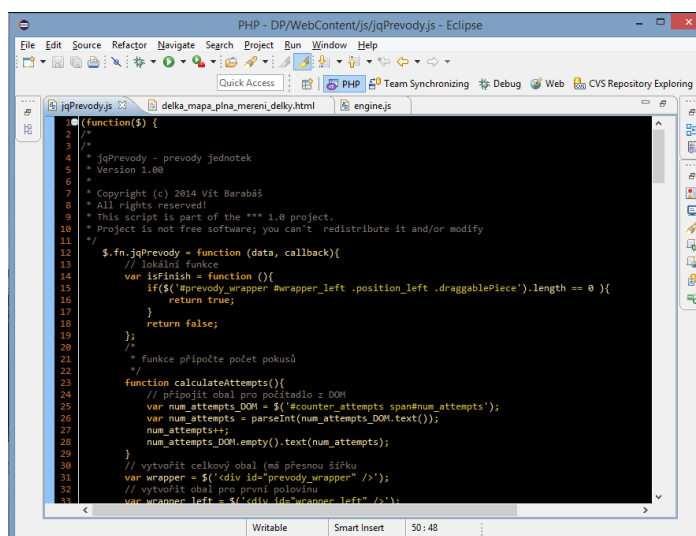
Takovýto příkaz vrátí ukazatel na data obsažená v poduzlu /data/variables/variable. Blíže bude datový model konkrétních úloh rozebrán v kapitole „*Návrh datového modelu úloh*“.

## 2.4 Použité vývojové nástroje

V této kapitole budou popsány vývojové nástroje, které byly použity při tvorbě této práce.

### 2.4.1 Eclipse

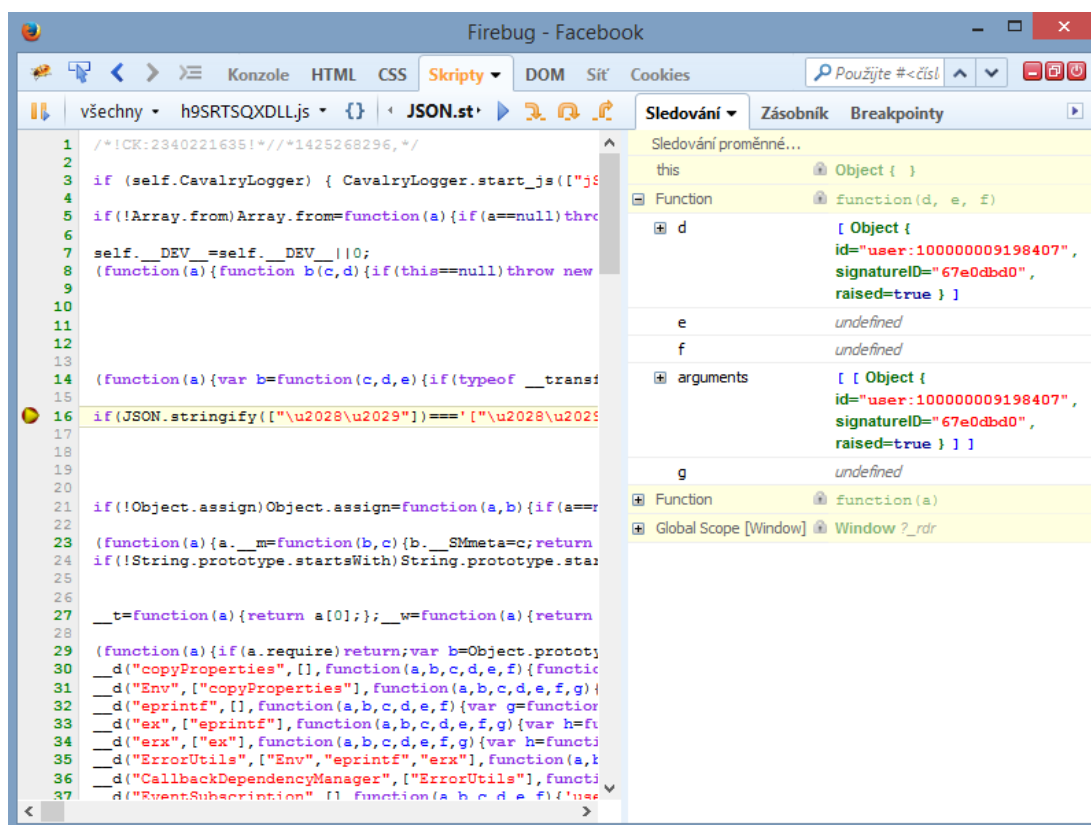
Pro psaní kódu bylo vybráno vývojové prostředí Eclipse. Důvodem, proč bylo vybráno právě toto prostředí, bylo kvalitní zvýraznění syntaxe a také cena. Eclipse je vývojové IDE naprogramované v jazyce Java a díky tomu je platformě nezávislé. Zároveň se jedná o volně dostupný nástroj šířený zdarma, do kterého lze doinstalovat celou řadu dalších rozšíření.



OBR. 2 VÝVOJOVÉ PROSTŘEDÍ ECLIPSE

## 2.4.2 Mozilla Firefox

Jako hlavní vývojový prohlížeč byl vybrán internetový prohlížeč Mozilla Firefox. Jsem na tento prohlížeč zvyklý a využívám v něm podpůrné vývojářské nástroje. Prvním doplňkem je HTML validátor, který slouží ke kontrole validity HTML. Dalším užitečným nástrojem pro odlazení Javascriptů je debugger Firebug. Tento doplněk umožňuje programátorovi zastavit běh programu na break pointu a prohlédnout si obsahy proměnných. Nejednou mi ušetřil čas i zdraví při hledání chyby.



OBR. 3 UKÁZKA ZASTAVENÉHO PROGRAMU

## 2.4.3 Verzování CVS

Protože se jedná o opravdu velký projekt, který trvá již déle než rok, bylo naprostou nutností zajistit bezpečnostní opatření, aby nedošlo ke ztrátě jednotlivých dat. Prvním zálohovacím opatřením samozřejmě bylo pravidelné zálohování kopie vývojové složky na externí disk a synchronizace do cloudu. Konkrétně za pomoci google drive aplikace. To ovšem neřeší problém s případnou ztrátou dat nechtěným přepsáním. Pokud bych si, dejme tomu, omylem přepsal již dříve vytvořený soubor nějakým jiným a nevšiml si toho včas, po synchronizaci by tato data byla ztracená, protože by se změny přepsaly i na zálohovaných zařízeních. Zároveň, bych ztratil přehled o

termínech změn. Tento problém elegantně řeší verzovací systémy, které jsou sice hlavně určené pro verzování projektů, na kterých pracuje více lidí najednou, ale pro mě je tento program hlavně zálohou jakékoli verze k datům verzování. Další velkou výhodou je snadná synchronizace zdrojových souborů na více počítačích, které bylo často využíváno. Na mém domácím linuxovém souborovém serveru byl nastaven verzovací server a tato složka se začala zálohovat do cloudu i na externí disk.

Výhodou verzovacího software CVS je v první řadě cena, protože jak server, tak klient je dostupný zcela zdarma. Verzovací server je dostupný k instalaci jako standardní balíček linuxové distribuce Ubuntu Server 14.02 LTS. Pro stažení zdrojového kódu bylo využíváno integrovaného modulu obsaženého přímo ve vývojovém prostředí Eclipse.

### 3 Test technologií na cílových zařízeních

Tato kapitola se věnuje testování podpory použitých technologií v mé bakalářské práci.

Testování bylo rozděleno do jednotlivých funkčních částí. Testují se převážně dva aspekty. Prvním testovaným aspektem je podpora napsaného programu na jednotlivých zařízeních. Zde se zaměřuji na to, aby se výsledná vygenerovaná aplikace zobrazovala co možná s nejmenšími rozdíly. Bohužel, to se často ukazuje jako neřešitelný problém, za který můžou v první řadě výrobci internetových prohlížečů. Tito výrobci často ignorují již standardizované funkce definované konsorciem *w3.org*. Některé problémy pak pramení z různého hardware, fungujícího často na rozdílných technologických principech.

Druhým ovšem neméně podstatným aspektem testování je zkouška viditelnosti na různých interaktivních tabulích. Cílovým kritériem je pro mě čitelnost textu na interaktivní tabuli ze vzdálenosti 8 m. Pokud je text z této vzdálenosti čitelný, pro tuto úlohu je toto kritérium splněno. Obecně se dá říci, že musím provést tuto kontrolu použitelnosti na každé nové úloze nezávisle na typu úlohy, protože je to otázkou obsahu. Dále se tedy budu v této kapitole omezovat pouze na první aspekt technické funkčnosti na daném hardware.

Při mých testech byla empiricky vyselektována jako minimální možná a stále ještě použitelná velikost fontů 28 px. Text psaný touto velikostí ovšem je opravdu na pomyslné hraně viditelnosti ze stanovené vzdálenosti 8 m a proto se snažím dodržovat rezervu velikosti fontů nastavených na hodnotu 30 px. Tato hodnota byla testována při výuce fyziky a to konkrétně dotazováním se žáků, jak danou úlohu vidí. Tyto odpovědi byly vždy poznamenány a porovnávány s vlastními zkušenostmi. Zde se ovšem projeví velké rozdíly mezi jednotlivými žáky. Zatímco někteří žáci byli schopni přečíst i texty vypsány fontem o velikostech 20 - 24 px, zhruba 30 % dětí nebylo schopno ze zadních lavic přečíst texty napsané menším fontem než zmiňovaných 28 px. Je ovšem potřeba říci, že děti, které nebyly schopné přečíst text ze vzdálenosti 8 m, tento text byly schopné přečíst z prvních lavic bez znatelných problémů. Proto tuto velikost fontu vnímám jako dostatečnou a žáky, kteří mají problémy se zrakem, lze řešit pouhým přesazením.



Druhým problémem, na který jsem při testování úloh narazil je použití podobných odstínů barev zvolených pro text i pozadí. Zde se může projevit porucha barvocitu, o které často žák ani učitel nemusí vědět. Následující tabulka vypovídá o rozšíření daltonismu. Je z ní patrné, že poruchou barvocitu trpí zhruba 8 % mužů a asi 0,5 % žen ([6], str 11).

	Zdroj dat	[Lukavský]	[Okabe a Ito]	[Webexhibits]		[Webvision]	
	Typ poruchy	Muži [%]	Muži [%]	Muži [%]	Ženy [%]	Muži [%]	Ženy [%]
Anomálie	Protanomálie	1,0	1,0	1,3	0,02	1,0	0,01
	Deuteranomálie	4,9	5,0	5,0	0,35	5,0	0,5
	Tritanomálie			0,0001	0,0001	vzácná	vzácná
Dichromazie	Protanopie	1,0	1,0	1,3	0,02	1,0	0,01
	Deuteranopie	1,10	1,0	1,2	0,01	1,5	0,01
	Tritanopie	0,001	0,001	0,0001	0,03	vzácná	vzácná
Monochromazie				0,00001	0,00001	vzácná	vzácná
Poruchy celkem		8,0	8,0	5,0-8,0	0,5	8,5	0,53

**OBR. 4 ROZŠÍŘENÍ DALTONISMU PODLE RŮZNÝCH ZDROJŮ ([6], STR. 11)**

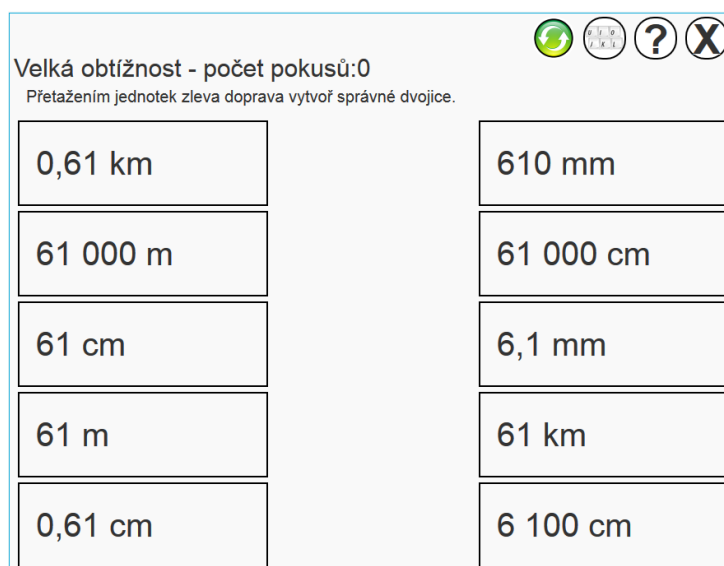
Jak lze z tabulky vyčíst, rozšíření různých úrovní poruchy barvocitu je celkem běžné a proto by s touto poruchou měly počítat i výukové materiály. Při návrhu vykreslení textu je snaha u jednotlivých úloh používat pouze černou barvu a naopak pro pozadí jednotlivých tvarů používat co nejvíce kontrastní barvy. Zde je ovšem velice těžké najít rovnováhu mezi funkčním, ovšem pro žáka nezajímavým barevným zobrazením a pěknou ovšem často přehlácanou a nepřehlednou aplikací. Osobně věřím, že by výukový materiál měl být především názorný a přehledný. U úloh, které testují žákovy vědomosti, jsou použity jednodušší grafické návrhy s pokud možno černobílým designem bez velkých efektů, naopak u motivačních úloh je používáno pro žáka zajímavějších barevných kombinací doplněných různými animacemi.

### 3.1 Testovací aplikace

Jako testovací aplikace jsou použity vzorové a naplněné učební materiály z kapitoly délka. Tato kapitola je v současnosti první a zároveň jedinou dokončenou kapitolou z celé učebnice. Konkrétně jsou vždy testované všechny typy jednotlivých úloh. Tak jak jsou postupně odhalovány nové chyby či nekompatibility, jsou úlohy opravovány a nové verze nahrávány na testovací i verzovací server.

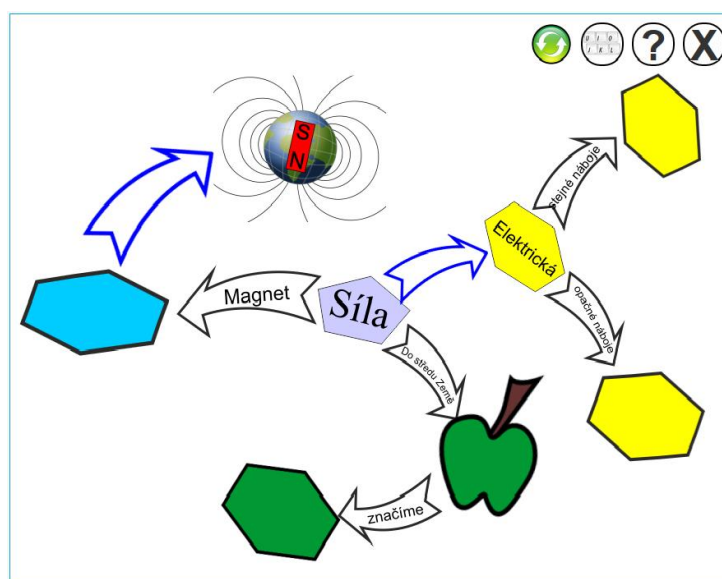
Testovacími aplikacemi byly zvoleny následující výukové metody. Pro testování podpory Drag and Drop API je použita aplikace „Převody jednotek“. Tato aplikace je

zvolena hlavně proto, že je zde testováno kromě samotného DND API i generování náhodných čísel. V této úloze se při každém zobrazení vygeneruje nová varianta převodů.



OBR. 5 TESTOVACÍ APLIKACE PŘEVODY JEDNOTEK

Další testovanou funkcionalitou je událost kliknutí či tapnutí (poťukat prstem) v případě použití dotykového zařízení. Pro testování této funkce byla vybrána úloha myšlenkové mapy, kde je testováno jak samotné klikání, tak i animování objektů ve scéně a uživatelský vstup z reálné či virtuální klávesnice. Tato úloha je poskytnuta i učitelům, kteří prováděli při své vlastní výuce pedagogickou sondu využitelnosti do mnou navržené myšlenkové mapy.



OBR. 6 TESTOVACÍ MYŠLENKOVÁ MAPA

## **3.2 Test podpory interaktivních tabulí**

Hlavním cílovým zobrazovacím zařízením pro výukové materiály jsou interaktivní tabule. Proto se v této kapitole převážně testuje toto koncové zařízení. Na českých základních školách jsou používány převážně 2 druhy interaktivních tabulí a to tabule Activboard a Smart Board.

### **3.2.1 Activboard**

Activboard je z těchto dvou zmiňovaných tabulí tou dražší, ovšem po technické stránce mnohem vyspělejší tabulí. Tyto tabule využívají pro zpětnou uživatelskou vazbu rezistivní dotykovou plochu a proto je pro ovládání tabule zapotřebí použití speciálního pera, které slouží jako ovládací nástroj. Z hlediska programování aplikace je pak toto pero stejným zařízením, jako je počítačová myš. Z hlediska kvality zobrazení mají projektory používané na interaktivních tabulích Activboard lepší svítivost, vyšší rozlišení a znatelně věrnější podání barev. Při testování úloh nebyl zjištěn žádný problém s podporou HTML5 či jQuery frameworku u tohoto typu zařízení.

### **3.2.2 Smart Board**

Interaktivní tabule Smart Board jsou levnější variantou pro školy, které nechtějí či nemohou investovat do jedné tabule bezmála 100 tisíc korun. Z hlediska programování aplikace je tato tabule poměrně problémová. Zpětná vazba je na této tabuli řešená ultrazvukovými senzory v rozích interaktivní tabule, které určují přiblížení se čehokoli k prostoru interaktivní tabule. Proto není potřeba se nutně tabule dotknout, aby byla vyvolána událost uživatelského vstupu. Výhodou této tabule je možnost ovládání aktivních prvků v aplikaci pouhou rukou, ovšem tato technologie s sebou nese i nepříjemnosti. Pokud je aktivní prvek příliš malý nebo je více aktivních prvků naměřeno příliš u sebe, tak vinou technologické nepřesnosti může docházet k chybným voláním uživatelského vstupu. Tento problém byl znát především u implementace virtuální klávesnice. Proto pokud navrhují univerzální aplikaci určenou k provozu na obou typech interaktivních tabulí, je nutné aplikace přizpůsobit fungování právě této tabuli. Jednoduše Activboard s tím co zvládne Smart Board nebude mít problém.

Druhým problémem, který byl při testování tohoto typu interaktivní tabule řešen, je odlišný způsob implementace uživatelského vstupu. Standardní událost pro táhnutí

bohužel na této tabuli nefunguje, a proto jsem musel hledat řešení tohoto problému. To bylo nakonec nalezeno v knihovně TouchPunch, jak bylo již zmiňováno v podkapitole 2.2.2. Tato knihovna nahrazuje a sjednocuje různé nestandardní uživatelské vstupy a přemalovává je na ty standardní.

Třetím problémem s tabulí Smart Board jsou použité projektoři, které jsou pravděpodobně limitovány nízkou cenou celého kompletu. Interaktivní tabule, které byly testovány, disponovaly projektorem s maximálním zobrazením 1024x768 obrazových bodů. Zároveň podání barev bylo vcelku tristní, což ovšem mohlo být způsobeno již velkým opotřebením testovaných zařízení. K dnešnímu dni jsou všechny typy úloh plně podporované na obou zde zmiňovaných tabulích v případě, že učitel používá moderní internetové prohlížeče, jak je vidět v tabulce přehledu podpory HTML5 funkcí.

### 3.2.3 Test podpory na desktopu

Test podpory aplikací na desktopu jsem do své práce zařadil jen pro úplnost, protože vzhledem k použité technologii, která přímo vychází z internetových stránek určených pro desktop, se neukázal žádný větší problém. Pravda, při vývoji byl asi největší problém s optimalizací aplikací pro internetové prohlížeče firmy Microsoft. Tento problém ovšem v drtivé většině vyřešilo použití Javascriptového frameworku jQuery, který tyto problémy automaticky odbourává.

Internetový prohlížeč	DND API	Generování náhodných čísel	Touch/Click	Animace
Google Chrome v. 41	✓	✓	✓	✓
Mozilla Firefox v. 36	✓	✓	✓	✓
Internet Explorer 11 (Desktop)	✓	✓	✓	✓
Internet Explorer 10	✓	✓	✓	✓
Internet Explorer 9	✗	✓	✗	✗

Podporováno  
Nepodporováno



### 3.2.4 Test podpory na mobilních zařízeních

Mobilní zařízení nebudou u prvního dílu těchto materiálů oficiálně podporované, ovšem do budoucna je zde jasně zřejmá tendence zapojovat tyto přenosné počítače do výuky na základních školách. Proto byl proveden test podpory navržených aplikací na jednotlivých mobilních platformách. Testování bylo provedeno na tabletu Apple iPad 2 mini retina s operačním systémem iOS 8.2, Android Prestigio PMT5777 s operačním systémem Android 4.4 a na tabletu Acer Aspire Switch 10,1“ s OS Windows 8.1. Jak můžete vidět z tabulky podpory mobilních zařízení, bez problémů lze v současnosti použít pouze tablety firmy Apple.

Zařízení	DND API	Generování náhodných čísel	Touch/Click	Animace
iPAD 2 mini retina	✓	✓	✓	✓
Android Prestigio PMT5777	✓	✓	✓	✗
Acer Aspire Switch	✗	✓	✗	✓

Podporováno  
Nepodporováno



Největší problémy byly s provozováním aplikací na tabletu od firmy Microsoft. Je opravdu obrovský rozdíl mezi internetovým prohlížečem Internet Explorer 11 pro desktop a internetovým prohlížečem Internet Explorer pro rozhraní Modern UI. Zatímco desktopová verze již HTML5 bez větších problémů zvládá a musím přiznat, že velice svižně oproti jiným současným internetovým prohlížečům, tak prohlížeč pro tablety je v době psaní této práce jeden obrovský problém. Tablet má velké problémy s rozlišením aktivního prvku a pokusu uživatele o posouvání obrazovky.

Na druhou stranu oproti tabletu s OS Android tento tablet neměl problém s plynulostí animování objektů. Tento problém ovšem osobně přisuzuji málo výkonnému procesoru MediaTek MT8382V. Z hlediska podpory technologie jako takové po stránce software není u tabletů s Androidem velký problém. Osobně přemýšlím, zdali má ovšem provozování HTML5 aplikací na zařízeních s operačním systémem Android, čili psaní univerzálních aplikací pro všechny zařízení, budoucnost. Tato platforma je natolik roztržštěná, tablety využívají takové rozličné druhy hardware,

s různým počtem operační paměti počínaje, přes diametrálně různě rychlé procesory, až po celou řadu nestandardních displejů disponujících různými rozlišeními konče. Osobně si v současnosti po mých zkušenostech s vývojem i testováním neumím představit aplikaci postavenou na technologickém základu s HTML5, která by si kladla za cíl plnou podporu všech zařízení s operačním systémem Android.

Naopak, vzhledem k určitému tlaku od firmy Microsoft na standardizaci a jistou unifikaci možného použitého hardware pro tablety s OS Windows, si podporu tohoto operačního systému v budoucnu představit dovedu. Zde ovšem potrvá určitou dobu, než tvůrci mobilního internetového prohlížeče dotáhnou podporu zpracování HTML5 na úroveň alespoň desktopové verze Internet Exploreru 11.

Jednoznačně nejdále jsou s připraveností na podporu technologie HTML5 produkty firmy Apple. To není velkým překvapením, protože počet produktů je oproti jiným platformám nejmenší a zároveň firma striktně dodržuje základní standardy jako poměr rozlišení u displejů svých tabletů. Zároveň tyto tablety již dnes disponují velice propracovaným internetovým prohlížečem Safari. Pokud bych měl za úkol vyvinout aplikaci pro mobilní zařízení, pak bych si v současnosti netroufl podporovat jiné zařízení než ty od firmy Apple.

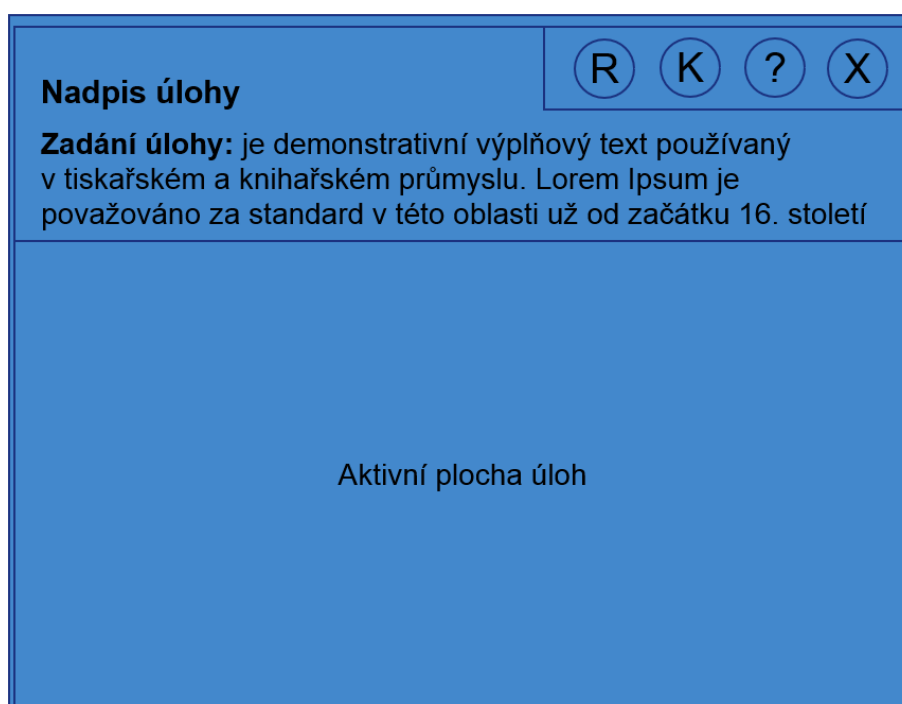
## 4 Vývoj aplikace

Tato kapitola se věnuje vývoji samotné aplikace. Popisované ukázky kódů nejsou samostatně funkční, avšak jsou v aplikaci použity.

### 4.1 Návrh úloh

Při návrhu grafické podoby jednotlivých úloh byly stanoveny základní obecné požadavky na úlohy.

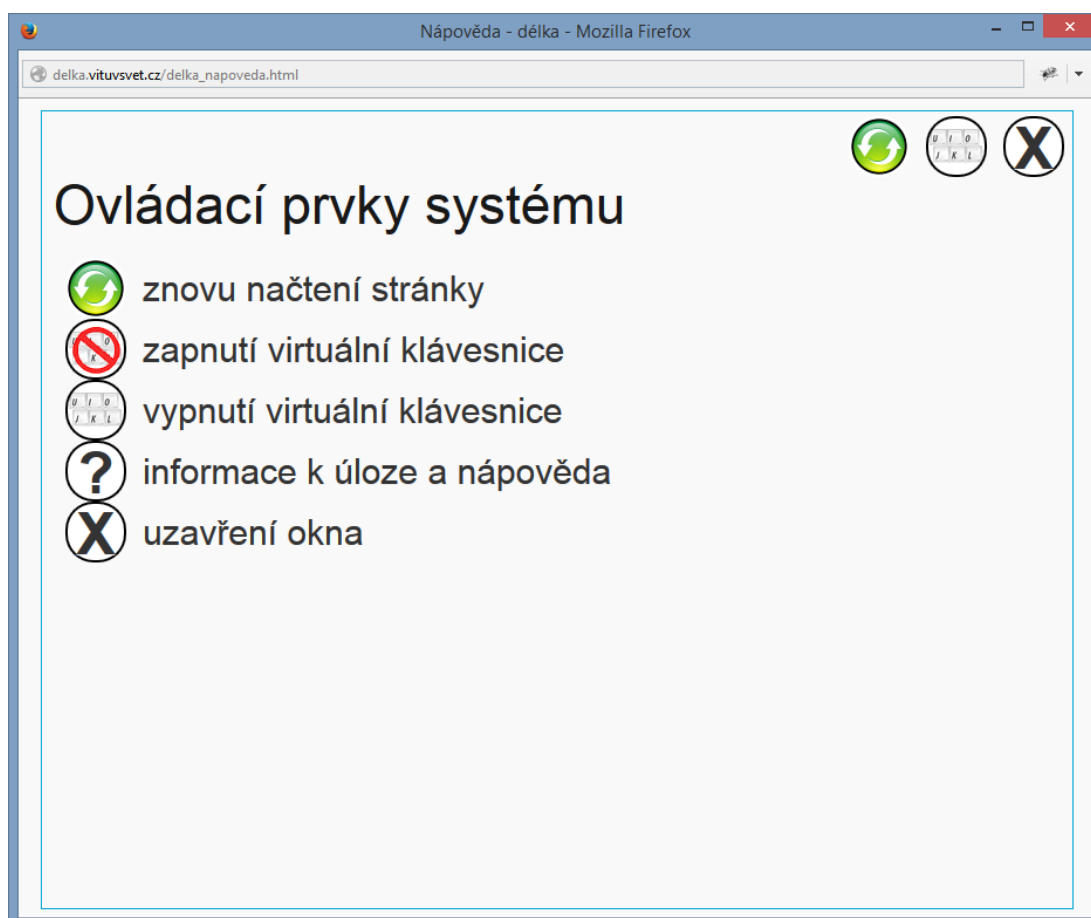
1. Velikost fontů musí být čitelná na interaktivní tabuli i ze zadních řad průměrné třídy.
2. Kombinace barev musí být viditelná a dostatečně kontrastní i pro děti s poruchou barvocitu.
3. Úlohy musejí být návrhem jednoduché a přehledné, aby se žák zaměřil na samotné učivo a ne na úlohu samotnou.
4. Všechny ovládací prvky, u kterých se předpokládá, že je budou na interaktivní tabuli ovládat žáci, musejí být umístěné do výšky tak, aby na ně bez problému dosáhli i menší žáci.
5. Po stranách jednotlivých úloh musí zůstat prázdný pruh, aby bylo možné rolováním ovládat úlohy na dotykových zařízeních.



OBR. 7 NÁVRH ROZVRŽENÍ AKTIVNÍCH ČÁSTÍ ÚLOH

Vzhledem k bodu č. 4 jsou všechny interaktivní části úloh orientované do spodních 2/3 pracovní plochy. Toto je důležité především u úloh určených pro nižší ročníky, kde se mohou vyskytovat žáci, kteří ještě nezačali růst a fyzicky do horních částí obrazovky nedosáhnou. Naopak do této horní 1/3 tabule se ukázalo jako vhodné umístění zadání úloh. Zadání úlohy by měli vidět i žáci v zadních lavicích a to co možná nejlépe. Pro zadání úloh bylo voleno větší fonty o velikosti minimálně 28 px. Pro rozvržení jednotlivých částí byl navrhnout drátěný model a otestován ze vzdálenosti na interaktivní tabuli. Protože vzniklé úlohy budou součástí komerčního produktu, byl návrh projednán s autory a zástupci nakladatelství. Po schválení drátěných modelů rozvržení komponent započal samotný konečný návrh a následný vývoj.

Poslední rozvrženou částí pracovní plochy je ovládací panel úloh, který obsahuje základní ovládací prvky, jako ukončení úlohy, aktivace nápovědy, přenačtení / restart úlohy a aktivace/deaktivace virtuální klávesnice. Tento panel musí být zobrazen vždy a je fixně orientován k pravému hornímu rohu nezávisle na šířce zobrazovacího zařízení.



**OBR. 8 UKÁZKA NÁPOVĚDY PROGRAMU VÝKOVÝCH MATERIÁLŮ PRO UČEBNICI FYZIKY 1**

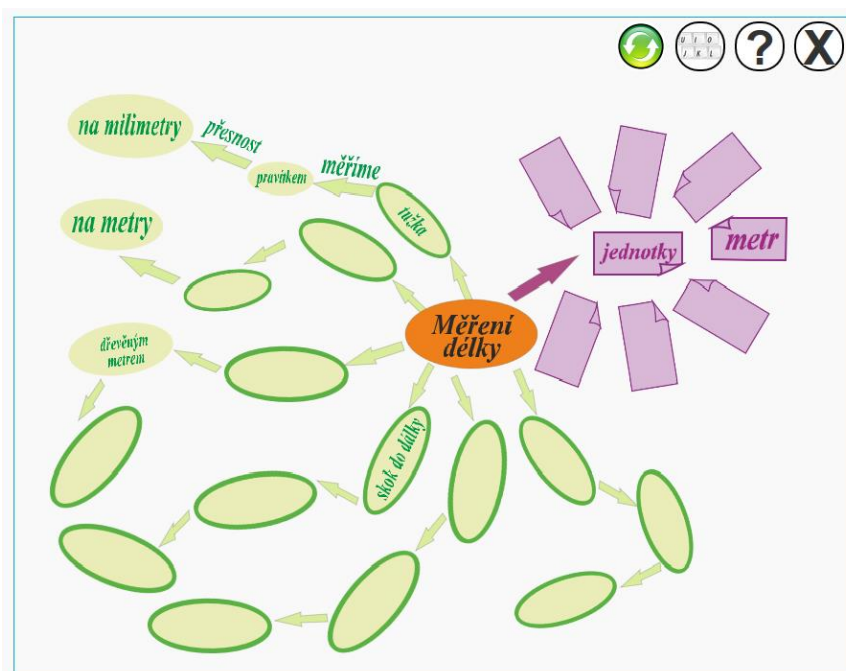


Všechny úlohy i ostatní části elektronické učebnice dodržují toto navržené rozvržení, pouze ovládací prvek pro zobrazení informací k úloze a nápovědy je volitelný. Některé typy úloh nepotřebují vždy zobrazovat další informace či nápovědu a tak lze tento ovládací prvek vypnout.

#### 4.1.1 Myšlenkové mapy

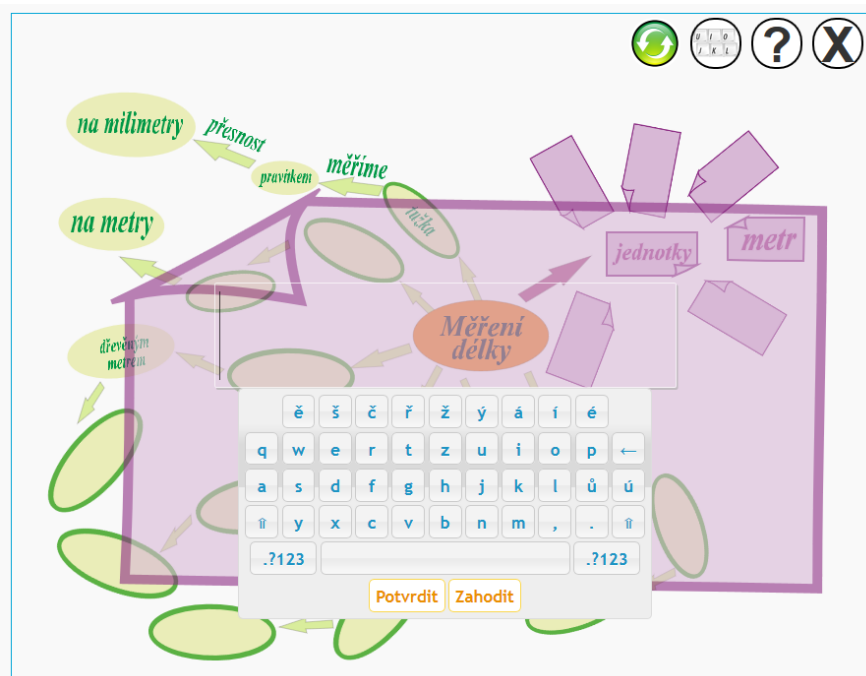
První navrženou úlohou byly myšlenkové či také pojmové mapy. Myšlenkové mapy jsou jedním z tzv. nástrojů vizualizace informací. Autorem pojmu mentální mapa je Tony Buzan. Podle něj se jedná o nástroj, který umožňuje lidskému mozku zapojit při zpracování informací obě mozkové hemisféry zároveň a podporuje asociativní uvažování. To má výrazně pozitivní vliv na to, jak si informace a znalosti zapamatujeme, znovu vybavíme nebo tvoříme nové. Pojmy mentální a myšlenková mapa jsou si v tomto modulu rovnocenné ([2]).

Myšlenkové mapy jsou dnes již obecně uznávanými nástroji při učení, kdy se žák neučí pouze izolovaným pojmem, ale učí se danou látku v souvislostech, relacích a vzájemné provázanosti. Já sám využívám myšlenkových map při běžném životě a mnohokrát se mi tento nástroj osvědčil při plánování či analyzování problému. Myšlenkové mapy byly použity při plánování vývoje mé bakalářské práce a použil jsem je i při plánování této diplomové práce. Tato myšlenková mapa je přiložena jako příloha.



OBR. 9 MYŠLENKOVÁ MAPA PŘED AKTIVACÍ

Myšlenkové mapy jsou typem úlohy, která využívá celou zobrazovací plochu. Pokud ovšem má být umožněno bezproblémové vyplňování aktivních částí i menším žákům, které jsou umístěné v horní třetině aktivní plochy interaktivní tabule, bylo potřeba navrhnout následující řešení tohoto problému. Jakmile se klikne do prostoru aktivní části myšlenkové mapy, bublina se plynule animovaně zvětší a odletí do prostoru mnou definovaných spodních 2/3 aktivní plochy, která byla stanovená jako povinná pro jednotlivé ovládací prvky. Aktivaci může provést učitel, ale dítě zůstává aktivní a hodnotu samo vyplní. Zároveň je zaručeno, že v případě použití virtuální klávesnice se tato klávesnice zobrazuje ve spodní třetině interaktivní tabule, čili v maximální výšce 1,2 m. Na obrázku č. 6 je ukázaná myšlenková mapa před kliknutím na aktivní bublinu.



**OBR. 10** MYŠLENKOVÁ MAPA PO KLIKnutí NA AKTIVNÍ BUBLINU

Výsledkem je vyplněná mapa, kterou může učitel žákům uložit či vytisknout jako studijní materiál. Na obrázku č. 10 je ukázán režim úlohy při kliknutí na aktivní část mapy se zapnutou virtuální klávesnicí. Vyplňovaný text je vždy umístěn uvnitř prostřední třetiny aktivní plochy a klávesnice je orientovaná do spodní třetiny aktivní plochy interaktivní tabule. Tím je zajištěno, že žák dosáhne na aktivní prvky, a ostatní žáci uvidí, co je do aktivní části vkládáno.

#### 4.1.2 Pexeso

Druhou navrženou úlohou je hra Pexeso. Tato hra je vcelku dobře použitelná v nižších ročnících, dokud jsou žáci ještě hraví. Této úlohy je běžně používáno v různých výukových materiálech. Tato hra má jednu velkou výhodu a to tu, že kromě učení látky samotné, hra také trénuje žákovu paměť. Zároveň pro žáka není tak náročné se soustředit na trénovanou látku delší dobu. Úloha je navržená jako dynamická úloha, pro kterou lze definovat počet hracích karet a tím regulovat obtížnost dané hry. V této hře, vzhledem k její povaze, je nutné využívat celou hrací plochu. Tuto hru bude pravděpodobně učitel spouštět na tabletu a tento obraz pouze posílat na interaktivní tabuli. Žáci pak hrají na tabletech každý svou vlastní hru a pouze na vyžádání odešlou obraz na interaktivní tabuli. Tato úloha je určena převážně pro žákovo opakování látky, a proto problém s rozvržením úlohy není zásadní.



OBR. 11 UKÁZKA HRY PEXESO

Standardně jsou pexesa navržena jako matice 3x4 obrázky, tedy 6 párů hracích karet. Pokud jsou vytvořené hry složeny z většího počtu hracích karet, průměrné žákovské řešení trvá již příliš dlouho a pro učitele tato motivační či trénovací úloha nebude k užítku. Na druhou stranu, pokud by chtěl dětem učitel nastavit vyšší obtížnost pro hru doma, lze nastavit různý počet hracích karet, který je omezen pouze velikostí aktivní plochy a v tomto důsledku pak velikostí samotné výsledné hrací karty. Dle

mých zkušeností na průměrném tabletu s úhlopříčkou 7“ není hra s více jak 20 ti kartami hratelná.

### 4.1.3 DND přiřazení párů

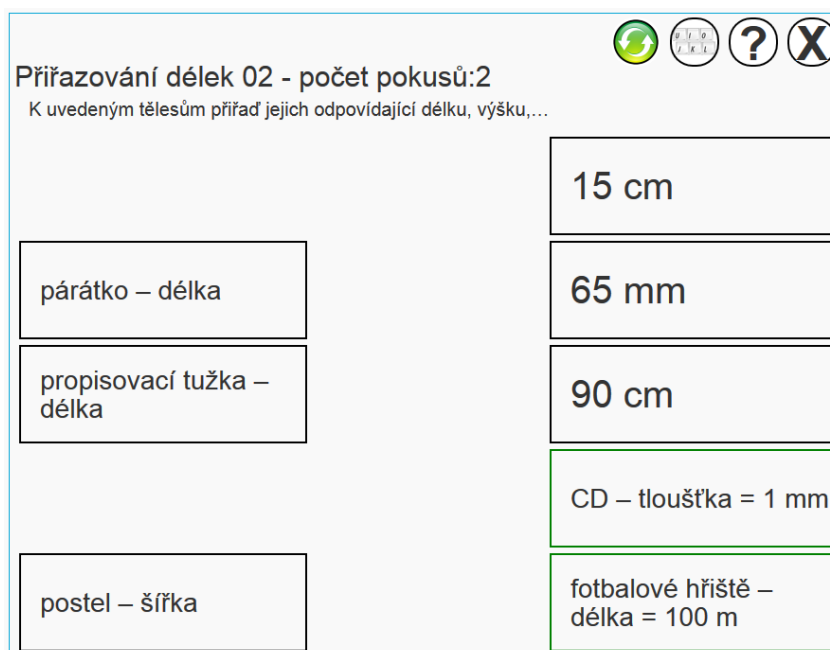
V pořadí třetí úloha „Drag and drop přiřazení párů“ je nástrojem pro opakování při hodině, tak i pro žákovo domácí přípravu. Tento nástroj generuje dvě skupiny aktivních bloků a žakovým úkolem je najít správné páry, které si odpovídají. Bloky na levé straně lze chytit a přetáhnout na pravé bloky.

S každým přetažením je pokus zaznamenán a v případě správného spojení odpovídajících párů, chycený blok zmizí z dostupného seznamu na levé straně a v pravé části se pozmění obsah přiřazeného bloku na kombinaci obou obsahů bloků oddělené znaménkem rovná se. V horní třetině aktivní plochy je umístěno počítadlo pokusů, kde je zaznamenán počet žakových pokusů o vyřešení převodu. Tato úloha má i další použití. Můžeme přiřazováním hledat páry textových informací, které si odpovídají, nebo například přiřazovat obrázky jejich názvům, přesností či jiným textovým popiskům. Posledním využitím je potom přiřazování obrázků k jiným obrázkům, kdy hledáme například páry stejného použití, jak můžete vidět na obrázcích 9 a 10.

Malá obtížnost - počet pokusů:1  
Přetažením jednotek zleva doprava vytvoř správné dvojice.

0,1 m	100 cm
0,1 km	1 000 m = 1 km
	10 cm
1 m	10 mm
1 cm	100 m

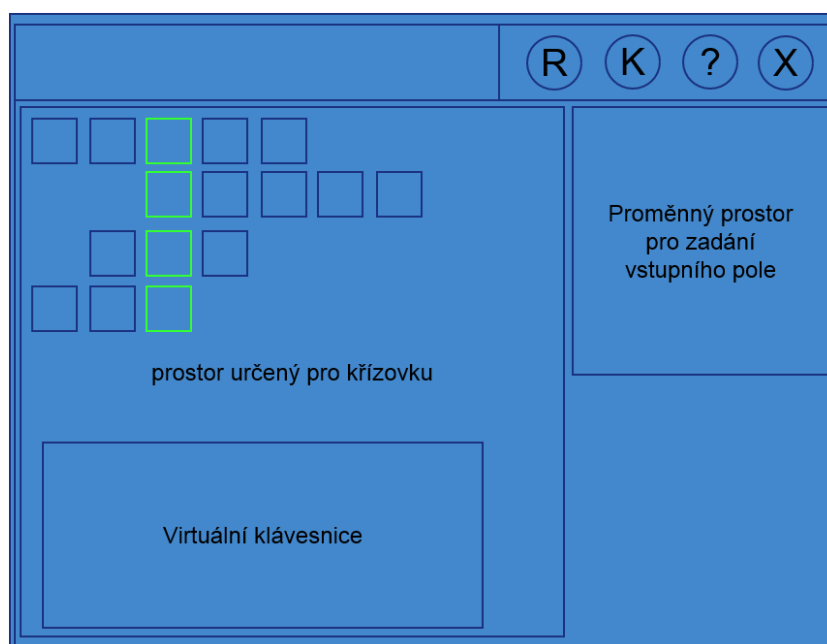
OBR. 12 ÚLOHA PŘIŘAZENÍ POUŽITÁ JAKO PŘEVODY JEDNOTEK



OBR. 13 DND PŘIŘAZOVÁNÍ PÁRŮ DÉLEK

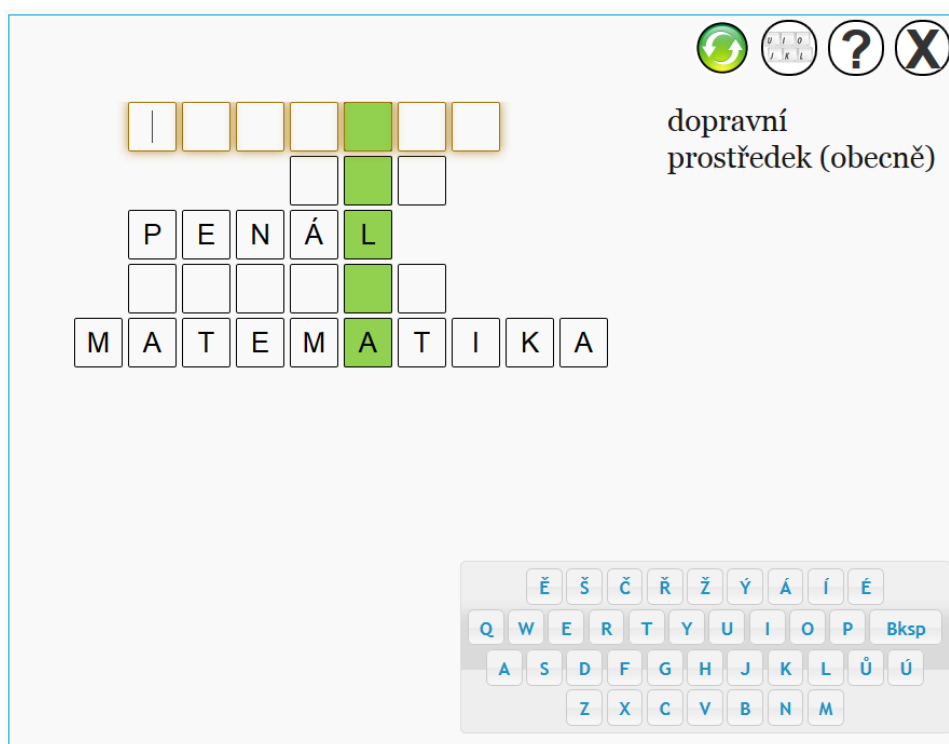
#### 4.1.4 Křížovka

Čtvrtou úlohou a v pořadí předposlední je úloha křížovka. Tato úloha je z hlediska návrhu asi nejsložitější, protože má využití, jak na interaktivní tabuli v rámci opakování či nějaké volnější hodiny, ale můžeme tuto úlohu nechat vypracovat žáky samostatně doma. Křížovka má zároveň trochu odlišné rozvržení. Potřebujeme využít co nejvíce prostoru na výšku pro skládání slov a tajenek. Proto pro tuto úlohu byl jako první odlazen návrh úlohy na drátěném modelu, jak můžete vidět na obrázku č. 11.



OBR. 14 DRÁTĚNÝ MODEL ROZVRŽENÍ ÚLOHY KŘÍŽOVKY

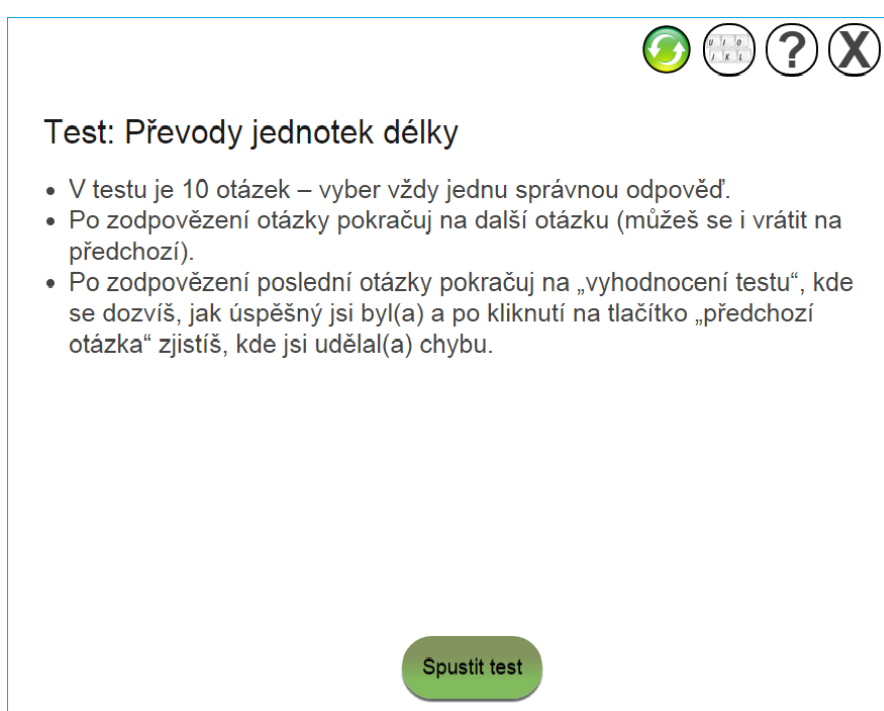
Jak si můžete všimnout, tak prvky které jsou aktivní, zároveň musejí být na interaktivní tabuli vidět. Tento problém byl vyřešen následovně. Křížovka samotná je umístěna co možná nejvýše, aby byla viditelná i ze zadních lavic. Samotné ovládání (vkládání písmen) je pak realizované virtuální klávesnicí umístěnou v dolní třetině pracovní plochy úlohy. Tímto je zajištěno jak vhodné zobrazení na tabuli či tabletu, tak i možné ovládání pro menší žáky. V pravé části pracovní plochy je sloupec zadání hledaného slova, jejíž obsah se mění po kliknutí na vyplňovaný řádek křížovky. Tímto je využit prostor zadání. Žádná použitelná slova nemohou být delší nežli 14 písmen, pokud slovo začíná na prvním sloupci. Tento parametr je samozřejmě svazující pro tvůrce křížovky, ovšem dle mých zkušeností získaných při plnění těchto křížovek, nebyl problém jednotlivé tajenky složit. Úloha používá pro vkládání písmen standardní vstupní formulářové pole. Výsledný prostor, který může autor křížovky použít je tvořen maticí 14x10 písmen. V rámci této matice si může tvůrce vytvořit jakoukoliv tajenku a má zaručené korektní i funkční zobrazení. Zároveň může autor předem vyplnit libovolná písmena v tajence a snížit tak obtížnost výsledné tajenky. Reálnou tajenku můžete vidět na obr. 15.



**OBR. 15 UKÁZKA POUŽITÍ KRÍŽOVKY NA INTERAKTIVNÍ TABULI**

#### 4.1.5 Test

Poslední navrženou úlohou v této diplomové práci je úloha pro testování. Použití je vcelku jasně dané. Kontrola žákových znalostí při hodině a žákova příprava na vyučování. U této úlohy je opět nutné zajistit jak dobrou viditelnost, tak vhodné umístění ovládacích prvků. Úloha samotná potom běží v celkem třech zobrazovacích režimech. Prvním režimem je režim zadání úloh. Zadání úloh má informativní charakter pro žáka. V zadání je popsán test, z kolika otázek se skládá, či zdali je nastaven časový limit. Informace jsou umístěné do horní třetiny aktivní plochy, zatímco spouštěcí tlačítko je umístěné úplně dole a zarovnané na střed.



OBR. 16 REŽIM ZADÁNÍ TESTU

Dalším režimem je zobrazení testové otázky. Každá testová otázka se zobrazuje samostatně. V prostoru, kde jsou zobrazené ovládací prvky, je umístěn objekt času. Tento objekt je schopen pracovat ve dvou režimech. První režim funguje jako odpočet určitého času. V úloze autor zvolí časový limit (v sekundách) za jaký musí žák test zvládnout. Po uplynutí této doby se test sám ukončí a přejde do režimu vyhodnocení. Žák zároveň průběžně vidí, kolik mu času zbývá. Druhým režimem nastavení objektu času je režim zobrazení aktuálního času. V tomto režimu nemá test časové omezení. Žák tak dokončí test až ve chvíli, kdy sám klikne při zobrazení poslední testové otázky na tlačítko „Vyhodnotit test“. Volitelně lze pro první režim nastavit upozornění žáka

na blížící se vypršení limitu blikáním odpočítaného času. Autor úlohy si může zvolit počet sekund před vypršením limitu, kdy začne objekt blikat.

00:04:45

10. Doplň následující převod:

55 m = 550 cm .

a) 550 mm

b) 550 cm

c) 0,055 km

d) 5,5 cm

Předchozí otázka

Vyhodnotit test

OBR. 17 UKÁZKA ŘEŠENÍ CHYBNĚ VYPLNĚNÉ ODPOVĚDI

Posledním zmiňovaným režimem testu je samotné vyhodnocení testu. V tomto režimu jsou zobrazeny statistiky úspěšnosti žakových odpovědí a žák se může vrátit tlačítkem „Předchozí otázka“ zpět. Postupně si může projít celý test a uvidí, kde udělal chybu. Správné odpovědi jsou podbarveny zeleně. Špatně vyplněné otázky jsou podbarveny oranžovou barvou, přičemž správná odpověď je podbarvená zeleně a orámovaná rudou přerušovanou čarou. Otázky které žák vůbec nevyřešil, jsou označeny jako správné odpovědi u špatně vyplněných odpovědí a to zeleným podbarvením s rudým přerušovaným orámováním.

V druhém režimu může zvolit autor mezi dvěma druhy otázek. Prvním typem je běžná otázka, kterou lze jen vybrat. Druhým typem otázky je doplňující otázka. Po výběru zvolené odpovědi se do podotázky doplní žákem vybraná odpověď. Tento typ otázek se přímo hodí pro různé doplnění vět, definic či například převodů.

Autor si může libovolně volit, z kolika možných odpovědí bude žák muset vybírat. Zároveň se vždy automaticky promíchává pořadí odpovědí. Díky tomu nemůžou žáci vyplňující test jednoduše opisovat podle pořadí a každý žák má trochu jiný test.



## 4.2 Návrh jádra aplikace

Co umožňuje systém je přímo definované tím jak je navržené jádro systému. Protože jsou v této práci navrhovány celkem nesourodé výukové metody, musel být návrh jádra aplikace co možná nejvíce univerzální a obecný. Základní myšlenkou mého návrhu je modulární rozdělení jednotlivých metod do samostatných javascriptových knihoven, které lze snadno načíst v hlavičce webové stránky konkrétního výukového materiálu.

### 4.2.1 Načítání XML

Jádro systému obstarává inicializaci systému a načtení základních konfigurací. Načítá systémové knihovny pro práci s XML a inicializuje používané Javascriptové doplňky třetích stran, které byly v práci využity. Příkladem navržené součásti jádra je metoda XML2JS.

```
jQuery.XML2JS = function (xml) {  
    var output = '';  
    // vytvoří kořen  
    var elementName = $(xml).prop('tagName');  
    var output =  
    $('<' + elementName + '></' + elementName + '>');  
    output.copyAttributes($(xml));  
    // uložit roota  
    var root = $(xml);  
    // načítat potomky z roota  
    var pom = $(root).prop('childNodes');  
    $(pom).each(function () {  
        var nodeNameChild = this.nodeName;  
        if(nodeNameChild == '#text'){  
            // text se pouze vloží  
            output.append($(this).text());  
        }  
    })  
}
```

```

        else if (nodeNameChild !== '#comment') {
            output.append(jQuery.XML2JS(this));
        });
    return output;
};

```

Účelem této metody je převedení dat ve formátu XML do vždy zpracovatelné podoby uvnitř javascriptu. Současné prohlížeče disponují parsery, které jsou schopné zpracovat XML soubory, každý výrobce to ovšem dělá trochu po svém a nemohlo by být zaručeno korektní načítání na většině koncových zařízeních. Proto byla napsána tato krátká metoda, která konverzuje v rámci rekurzivního průchodu stromové struktury XML dokumentu celý dokument a vytvoří z něj odpovídající stromovou strukturu dat naformátovaných do podoby Javascriptových objektů.

Tato metoda postupně prochází celou XML strukturu a vytváří vlastně kopii jen jinak naformátovanou. Tím se v daný okamžik zdvojnásobují paměťové nároky aplikace. Tento nešvar ovšem není zásadní, protože jsou v drtivé většině případů načítány pouze konfigurace jednotlivých úloh. XML samotné ovšem umožňuje uchování i binárních dat uvnitř obsahů jednotlivých elementů. Pokud by například autor úlohy uložil uvnitř XML dokumentu zakódovaný obrázek ve formátu Base64, což je formát pro uložení obrázků v textové podobě, systém tento obrázek bez problémů načte a dále zpracuje, jen bude obsažen v operační paměti ve chvíli dokončení konverze dvakrát. Mnohem lepším použitím je ukládání obrázků mimo datové soubory jednotlivých úloh v textové podobě a pouze načítat URL adresy k jednotlivým obrázkům. Jeden takto uložený obrázek pak může být načten ve více úlohách. Pokud je XML soubor převeden do JS struktury objektů, můžeme původně načtený dokument uvolnit z paměti.

## 4.2.2 Generování náhodných čísel

Další mnou navrženou a nutno říci často využívanou metodou jádra je funkce `randomInt`.

```
jQuery.randomInt = function (max, min)
{
    if(typeof min == 'undefined'){min = 0;}

    return Math.floor(Math.random() * (max-
min+1)+min)
};
```

Tato metoda využívá již předpřipravených matematických metod objektu `Math` a pouze umožňuje definovat výskyt generovaných čísel v považovaném intervalu.

## 4.2.3 Mazání prvků z pole

Protože jazyk Javascript v základu neumožňuje dynamicky uvolňovat jednotlivé prvky pole, rozšířil jsem standardní JS objekt `Array` o metodu `remove`.

```
Array.prototype.remove = function (el) {
    return this.splice(el, 1);
};
```

K této funkci bylo využito metody `splice`, která je v současnosti již velice dobře podporovaná. Jak můžete vidět z tabulky „Podpora metody v internetových prohlížečích“, metoda je podporovaná defakto všemi současnými prohlížeči ([7]).

Method	Google Chrome	Internet Explorer	Mozilla Firefox	Safari	Opera
splice()	1.0	5.5	1.0	Yes	Yes

Tabulka: Podpora metody `splice()` v internetových prohlížečích<sup>8</sup>

Metoda je velice rychlá pro použití u malých polí, ovšem velmi ztrácí na použitelnosti při zpracování rozsáhlých polí. V případě odebrání jednoho prvku, je vytvářeno druhé nové pole bez použití odebíraného prvku pole. To je samozřejmě jak časově, tak i paměťově velice náročné.

<sup>8</sup> Tabulka dostupná z adresy: [http://www.w3schools.com/jsref/jsref\\_splice.asp](http://www.w3schools.com/jsref/jsref_splice.asp)

#### 4.2.4 Kopírování atributů

Další metodou, kterou bylo rozšířeno jádro systému je metoda `copyAttributes`.

```
$.fn.copyAttributes = function(object) {  
    // nastavit rodičovsky element  
    var parent = $(this);  
    // překopírování nenullových atributů  
    $.each($(object).prop('attributes'), function() {  
        parent.attr(this.name, this.value);  
    });  
};
```

Tato metoda umožňuje snadné zkopírování všech atributů z libovolného elementu ve prospěch elementu, na kterém byla metoda zavolaná. Zdrojový element, ze kterého se budou atributy kopírovat, je předán metodě v podobě ukazatele na objekt v prvním parametru metody.

#### 4.2.5 Automatické přetypování proměnných

Při programování aplikace se vyskytl jeden problém. Ve chvíli kdy byl zpracováván např. uživatelský vstup z formulářových textových vstupů a jako hodnotu předával logickou hodnotu `true` nebo `false`, některé prohlížeče tuto hodnotu vnímají jako řetězec. Jazyk Javascript je jazyk, který automaticky přetypovává proměnné, což ovšem neznamená, že by proměnná v jazyce javascript neměla datový typ. O tomto jsem se celkem jasně přesvědčil ve chvíli, kdy byla odeslána formulářovým vstupem typu `hidden` hodnota `false`. Tato získaná proměnná byla použita jako parametr podmínky v běžícím kódu a kód byl na některých prohlížečích jako Internet Explorer 9 vnímán jako pravda. Tyto prohlížeče mají implementovaný příkaz podmínky jako vstup, který je typu `string` (řetězec) a má nenulovou délku je vždy vyhodnocen jako pravda. To samé platí pro vstup v podobě čísel, kdy `0` je vnímána jako logická nepravda a cokoliv jiného jako logická pravda, ovšem pouze do chvíle než je tato hodnota přetypována do typu řetězce. Pak je tato hodnota `0` vnímána jako řetězec nula a tím pádem řetězec s nenulovou délkou jako logická pravda.

Při hledání důvodu tohoto pro mě zvláštního chování, bylo zjištěno, že některé prohlížeče provádějí automatické přetypování ve chvíli, kdy použijeme proměnnou

jako součást jiného spojeného řetězce. Protože je nutné vždy zajistit správné chápání obsahu proměnné, byla napsána rozšiřující funkce `parseBool`.

```
jQuery.parseBool = function (value) {  
    switch(jQuery.type(value)) {  
        case 'boolean':  
            return value;  
            break;  
        case 'string':  
            if(value == 'true') {  
                return true;  
            }  
    }  
    return false;  
};
```

Tato metoda zaručuje vždy správné chápání obsahu proměnné. Zároveň slouží jako jistá forma ochrany uživatelského vstupu, protože cokoliv neočekávaného je konvertováno jako logická hodnota `false`.

#### 4.2.6 Vkládání výukových metod

Jádro aplikace obsahuje načítací metody pro vkládání výukových metod do scény. Jednotlivé metody se jmenují `addPexeso`, `addSlepaMapa`, `addPrevody`, `addKrizovka` a `addTest`. Názvy metod jsou vcelku sebe vysvětlující, a proto nebude vysvětlováno, co každá z nich přidá do scény. Naopak budou popsány jednotné principy používání této skupiny metod jádra systému. Každá z metod obsahuje v prvním parametru ukazatel na XML data definující podobu této konkrétní metody. Pokud má úloha očekávané řešení, jako například nalezení všech párů karet u metody `pexeso`, druhý parametr obsahuje ukazatel na anonymní funkci nebo také `callback`, která bude provedena po dokončení dané úlohy. Tím je umožněno autorovi výukových materiálů nastavovat, co se má stát po vyřešení úlohy. Této funkcionality je využíváno při řetězení úloh viz kapitola *Řetězení úloh*.

#### 4.2.7 Načítání výukových úloh

Aby bylo sjednoceno a zabstraktněno načítání, byla navržena jedna mezivrstva mezi načítáním konkrétního typu úloh a rozhodovacím voláním metod. To bylo

zaobaleno do metody loadApp. Tato metoda provádí asynchronní načtení datového XML souboru definujícího výukovou úlohu za pomoci technologie AJAX (Asynchronous JavaScript and XML). V tomto datovém souboru je uložena informace o typu načítané úlohy (blíže popsané v kapitole 4.3), a automaticky se zvolí správná metoda k provedení načtení této úlohy.

```
$.fn.loadApp = function (XML_link, onLoad, onFinish){  
    var parent = $(this);  
    $.ajax({  
        type: 'GET',  
        url: XML_link,  
        cache: false,  
        dataType: 'xml'  
    }).done(function (xml_data){  
        var type = $(xml_data).find('app type');  
        var type_t = type.text();  
        jQuery.help_show_id = $(xml_data).find('data  
helps_data default_help').text();  
        jQuery.default_help_id = jQuery.help_show_id;  
        // načíst nápovědu  
        xml_js =  
jQuery.XML2JS($(xml_data).find('app'));  
        jQuery.helps_data = xml_js.find('helps_data');  
        switch(type.text()){  
            case 'test':  
  
        parent.addTest($(xml_data).find('app'), onLoad,  
onFinish);  
  
            break;  
            case 'krizovka':  
  
        parent.addKrizovka($(xml_data).find('app'), onLoad);  
  
            break;  
            case 'prevody':  
  
        parent.addPrevody($(xml_data).find('app'), onLoad);
```

```

                break;
            case 'mapa':
                // načíst hru

parent.addSlepaMapa ($(xml_data).find('app'));

                onLoad();

                break;
            case 'pexeso':
                // načíst nápovědu

parent.addPexeso ($(xml_data).find('app'), onLoad);

                break;
            default:
                alert('Aplikace nenalezena!');
        }
    }).fail(function (jqXHR, textStatus){
        alert('Chyba: ' + textStatus);
    });
};

```

Jak můžete vidět v kódu, tak při volání této metody se automaticky převedou načtená data ve formátu XML do JS objektové struktury. V případě, že dojde k chybě při načítání zdrojového souboru, který může být např. nedostupný či dojde k výpadku spojení před úplným stažením datového souboru, tak systém tento problém zachytí a vypíše chybovou hlášku s popisem této situace.

#### 4.2.8 Výpis dialogů

Další užitečnou funkcí systému je snadné vypisování dialogů. Dialog je z technického hlediska pouhý absolutně pozicovaný div element, umístěný na střed zobrazovací plochy, orámovaný a v pravém horním rohu mu je přidáné zavírací tlačítko. Toto tlačítko vymaže tento blok ze scény. Nastavitelně lze tento dialog udělat aktivní pro přetahování po scéně. Dialog lze přidat do scény voláním metody addDialog

```

$.fn.addDialog = function (content, width, height, id,
callbackExit, draggable)

```

Prvním parametrem je vkládaný obsah ve formátu javascriptové stromové strukturu a vloženo může být libovolné HTML. Parametry `width` a `height` definují rozměry budoucího dialogu. Parametr `id` obsahuje hodnotu identifikátoru, která bude nastavená pro hlavní element dialogu a lze ho použít k unikátnímu popisu tohoto dialogu. V závislosti na nastavení tohoto parametru, se mohou na výsledný dialog aplikovat různé styly a autor úloh tak může měnit vzhled výsledného dialogu. Parametr `callbackExit` je ukazatel na zpětné volání funkce, které vykoná anonymní funkci po vypnutí tohoto dialogu. Tato funkce slouží např. k automatickému uložení posledních výsledků či stavu před uzavřením. Poslední parametr `draggable` je parametr obsahující hodnoty `true` a `false`. Tímto parametrem uživatel povoluje citlivost na táhnutí dialogu ve scéně.

#### 4.2.9 Dialog s odkazy na další úlohy

Celkem častým použitím dialogů je jejich použití jako rozcestníků. Proto byla do jádra zakomponována již předpřipravená metoda `addDialogWithHTMLHrefs`, která načte z JS objektových dat cílové odkazy na další úlohy a vygeneruje naplněný dialog.

```
$.fn.addDialogWithHTMLHrefs = function (data, width, height, id, closeWindow)
```

Parametr `data` je ukazatel na data, ze kterých se budou generované linky načítat. Parametry `width` a `height` opět nastavují výsledný rozměr dialogu a parametr `id` i `closeWindow` má stejnou funkci jako u běžného dialogu. Generované objekty musejí být uloženy uvnitř elementu `data_link` a třídou `dialog_click`. Z každého takového elementu bude vygenerovaný jeden odkaz na další úlohu. Důležité je, že tyto odkazované úlohy se začnou asynchronně načítat do paměti až v okamžiku uživatelského kliknutí. Tím je zajištěno rychlé načítání dialogů.

#### 4.2.10 Zobrazení nápovědy

Celý systém je od samého začátku navržený tak, aby se co nejméně tříštily jednotlivé části. Proto je v rámci celého systému nastavená jednotná práce s nástrojem nápověda. V rámci úlohy se nastavuje systémová proměnná `msg_box_is_hidden` a v závislosti na ní, se buď zobrazuje či nezobrazuje dialog nápovědy. Pro nápovědu je vždy použito jedno stejné okno, jen se mění v průběhu běhu programu obsah nápovědy. Obsah této nápovědy se automaticky načítá v rámci metody `loadApp` a mění se pouze parametr ukazující na aktuálně se zobrazující obsah nápovědy `help_show_id`. Samotné vyvolání nápovědy je pak realizováno metodou `showHelpBox`.



#### **4.2.11 Nastavení virtuální klávesnice**

Pro vygenerování virtuální klávesnice byl použit již hotový veřejně dostupný opensource plugin Virtual Keyboard<sup>9</sup>. Tento plugin dovoluje celou řadu přizpůsobení a je velice snadno implementovatelný do projektů. Programátor musí pouze nakonfigurovat požadovaný vzhled virtuální klávesnice, musí definovat klávesy obsažené na klávesnici a pak jen toto nastavení přiřadí objektu `makeKeyboard`. U takto definovatelné klávesnice je v plné režii programátora, jak bude tato klávesnice vypadat. V aplikaci je dodržováno standardní rozvržení QWERTZ, na které jsou žáci zvyklí u počítačových klávesnic.

#### **4.2.12 Generování ovládacího panelu**

Protože je u každé úlohy generován ovládací panel aplikace, obsahující tlačítko pro vypnutí úlohy, nápovědu, ovládání virtuální klávesnice a načtení úlohy. Do jádra byla přidána metoda `makeControlPanel`, která přidá do bloku pro umístění ovládacích tlačítek jednotlivá tlačítka a namapuje na ně reakce.

### **4.3 Návrh datového modelu úloh**

Velice důležitou částí práce je návrh datového modelu jednotlivých úloh. Tento datový model specifikuje strukturu dat, která poté může být automatizovaně zpracovávána. Základní myšlenkou navrženého datového modelu je, co možná nejvíce zachovat unifikovaný zápis úloh. Pokud se v dané úloze něco opakuje, pak to musí být zapsáno stejně. Tímto je programátor schopen znovu využívat stejné části kódu a v případě nalezení problému, pak opravit chybu jen na jednom místě.

#### **4.3.1 Základní struktura datových souborů aplikací**

Základem každé úlohy je kořenový element `app`. Tento element je pro všechny úlohy povinný, nemá žádný volitelný atribut a slouží jako kořen stromové struktury. Potomky kořenového elementu jsou elementy typu definující typ načítané úlohy, element `background` definující pozadí aplikace, `helps_data` obsahující jednotlivé obsahy pro nástroj nápověda z ovládacího panelu.

---

<sup>9</sup> Plugin Virtual Keyboard je dostupný z adresy: <http://mottie.github.io/Keyboard/>

### 4.3.2 Datový model nápovědy

Datová struktura nápovědy je opět společná pro všechny úlohy. Některé úlohy tuto nápovědu průběžně mění, jiné mají obsah nápovědy po celou dobu stejný. Prvním potomkem elementu *helps\_data*, je element *default\_help*, který obsahuje identifikátor defaultního obsahu nápovědy. Dále následuje libovolný počet elementů *help*. Každý element *help* povinně obsahuje atribut *id* sloužící jako identifikátor, který musí být unikátní v rámci celého datového souboru.

Výplň jednotlivých obsahů pro nápovědy může být libovolné HTML. Výsledné HTML bude stylované platnými a načtenými CSS styly, čili forma nápovědy je v plné režii autora výukového materiálu. Pokud autor bude chtít dát žákovi jako nápovědu video, nebrání tomu žádná technická překážka.

Různé úlohy s touto logikou přímo spolupracují. Jako příklad můžu uvést úlohu slepé mapy. V této úloze jsou definované aktivní bubliny (prvky) elementem *bublina*. Pokud autor jednotlivým elementům nastaví atribut *id*, tak se tyto aktivní prvky napojí na shodné nápovědy a v případě aktivace tohoto prvku se nastaví odpovídající obsah nápovědy. Stejně je to v případě spouštění více aplikací v jednom aktivním okně za pomoci vnořených dialogů. Pokud je aktivní dialog s načtenou úlohou, automaticky se v tomto okně nastaví nápověda z datového souboru aktivní úlohy.

### 4.3.3 Datový model metody křížovka

Každá úloha obsahuje některé unikátní nastavení reprezentované vlastními elementy. Příkladem je úloha křížovka. Kromě již zmiňovaných elementů obsahuje kořenový element *app* i potomky v podobě konfiguračních elementů *num\_cols* a *num\_rows*, které slouží pro nastavení generování matice vstupních polí. Dále je zde obsažena trojice upřesňujících parametrů *letter\_width* a *letter\_height* definující rozměry zobrazovaných vstupních polí pro vkládání písmen. Posledním, z trojice nastavujících elementů je element *margin*, který definuje velikost okraje udávaný v jednotkách pixel. Samotná tajenka je pak definována elementem *words* obalujícím všechna definovaná slova křížovky. Uvnitř tohoto elementu je libovolný počet elementů *word* s atributem *value* obsahujícím očekávanou vyplněnou hodnotu a atributem *help* který ukazuje na hodnotu obsahu, který se má při vybrání daného slova doplnit do nápovědy nástroje nápověda v ovládacím panelu.

Jednotlivá aktivní slova jsou definovaná uvnitř elementu *letter\_positions* obsahujícím jednotlivá písmena *letter\_position*. Tento element má povinné atributy *position\_x* a *position\_y*, které určují souřadnice uvnitř zobrazovací matice vygenerované dle parametrů *num\_cols* a *num\_rows*. U tajenky je pak nutné vyplnit pro element *word* atribut *value* hodnotou TAJENKA. Tím se nastaví slovu jiné barevné orámování. Pro každý element *word* je definován subelement *help\_image* tento element obsahuje html, které se zobrazí v hádance daného slova a bude se měnit s výběrem jednotlivých slov.

#### 4.3.4 Datový model metody DND přiřazení párů

U tohoto modelu jsou 2 možné způsoby použití. První způsobem použití je generování převodů jednotek. Volitelně může autor použít element *variables*. Uvnitř tohoto elementu může být libovolný počet elementů *variable* symbolizujících proměnné. Atributy *interval\_start* a *interval\_end* nastavují interval, ze kterého se bude generovat náhodné číslo. Takto lze definovat jak náhodné intervaly, tak konkrétní číselné hodnoty. Hodnotou tohoto atributu jsou definované řetězce, které se budou nahrazovat uvnitř doplňovaných textů. Element *transfers* má jeden povinný atribut *select* definující počet vybraných převodů z uživatelem definovaných a možných k použití. Jak jsem již naznačil, uvnitř tohoto elementu může být definovaný libovolný počet převodů reprezentovaných elementem *transfer*. Tyto elementy mají povinné atributy *used\_variable* odkazující na použité proměnné v tomto převodu. Pokud využívá převod více jak jednu proměnnou, proměnné jsou oddělené separačním znakem středník. Identifikátoru odkazující na tyto proměnné jsou navázány na atribut *var\_id* elementu *variable*. Tyto proměnné budou nahrazovány hledanými podřetězci uvnitř atributů *left\_value* a *right\_value*. Volitelně pak lze ovlivňovat velikost vykreslovaných fontů atributy *end\_font\_size* a *start\_font\_size\_left* či *start\_font\_size\_right*.

Obdobným způsobem jako u převodů se pak vytváří jakékoliv převody, jen není potřeba v datovém souboru definovat proměnné.

#### 4.3.5 Datový model metody slepá mapa

Úloha slepá mapa obsahuje mimo společných subelementů kořenového elementu *app* i definici aktivních prvků. Ty jsou reprezentovány elementem *bublina* a obsahují

celou řadu atributů. Atributy jsou jen přeložené do angličtiny a je vcelku jasné co dělají:

<ul style="list-style-type: none"><li>• data-start_scale</li><li>• data-anim_scale</li><li>• data-start_rot</li><li>• data-anim_rot</li></ul>	<ul style="list-style-type: none"><li>• data-start_pos_top</li><li>• data-anim_pos_top</li><li>• data-start_pos_left</li><li>• data-anim_pos_left</li></ul>
---	---

Obecně jde o nastavující parametry vzhledu aktivních prvků slepých map. Autor mapy má maximální volnost při tvorbě a v závislosti na nastavení parametrů může vytvářet různé efekty jako animované vrtule. Pokud například nastaví autor počáteční hodnotu rotování objektu uvnitř scény data-start\_rot na -360 a cílovou hodnotu rotace data-anim\_rot na 360. Objekt bude před kliknutím nastaven bez natočení a po kliknutí se 2x otočí. Takto lze kombinovat všechny parametry a prolnout tak více druhů animací v jeden okamžik. Každý objekt pak může být animovaný samostatně.

#### 4.3.6 Datový model metody Test

Předposlední úlohou, jejíž datový model zmíním, je úloha test. Tato úloha mimo společných elementů uvnitř kořenového adresáře má i element *begin\_test\_info* sloužící pro definice obsahu zobrazenému na začátku testu. Obsahem může být opět libovolné HTML. Druhým nastavujícím elementem je element *timer* definující nastavení objektu času. Tento objekt je schopen pracovat ve dvou režimech. První režim pouze ukazuje aktuální systémový čas na daném počítači a autor úlohy ho nastaví hodnotou *time*. Naopak režim odpočítávání se nastaví hodnotou atributu *type* nastavenou na *countdown*. Tento režim pak začne odpočítávat dobu nastavenou uvnitř elementu *time\_in\_seconds*. Volitelně pak lze nastavit element *time\_when\_start\_blicking\_in\_s* reprezentující dobu, po kterou před vypršením časového limitu začne objekt na obrazovce blikat.

Testovací otázky jsou obaleny elementem *test\_questions* uvnitř něj může být libovolný počet testových otázek reprezentovaný elementem *test\_question*. Uvnitř tohoto elementu jsou dva povinné elementy. Prvním elementem je element *question*,

který obsahuje subelement *title*. Uvnitř tohoto elementu je uložena otázka, na kterou se autor ptá. Volitelně lze použít element *subtitle*, který může obsahovat doplnění otázky a v případě zvolení typu otázky *abc\_complement* je tento element povinný. Jak bylo již naznačeno, v návrhu aplikace *test*, otázky mohou být použity dvojího typu. První typ *abc* otázek je běžná otázka s výběrem jednoho řešení. Druhý typ otázek jsou doplňující otázky, které vybranou hodnotu doplňují do hodnoty uložené v elementu *subtitle*. Hodnota uložená uvnitř tohoto elementu musí obsahovat řetězec *{x}*, který bude nahrazen vybraným řešením.

Uvnitř elementu *test\_question* je povinný element *answers*. Tento element obsahuje libovolný počet otázek reprezentovaných elementy *a*. Odpověď, která je označena jako správná, má vyplněný atribut *is\_true* na hodnotu *true*.

#### **4.3.7 Datový model metody Pexeso**

Posledním datovým modelem, který bude popsán, je metoda *pexeso*. Tento model kromě povinných atributů obsahuje element *settings*, který obsahuje nastavení doby v sekundách, po kterou zůstane karta otočená. Jednotlivé karty jsou definované nepárovým elementem *karta*, který obsahuje povinné atributy *data-text* a *src*. První atribut obsahuje textovou část pro každý pár, zatímco atribut *src* obsahuje odkaz na obrázek použitý u druhé poloviny karet.

## 4.4 Řetězení úloh

Již od samého začátku jsou navrženy úlohy pro případné řetězení jednotlivých úloh za sebou. Je ovšem potřeba říci, že je možné řetězit úlohy, u kterých lze očekávat dokončení. Pokud v nějaké úloze řetězím např. po dokončení hry pexeso slepou mapu použitou jako popis jednotlivých obrazovek použitých v pexesu, dále již nemůže být tato úloha dále propojovaná, protože metoda slepé mapy nemá definované dokončení.

Samotné řetězení více úloh za sebou je řešeno v inicializačním scriptu při volání metody `loadApp`. V následující ukázce kódu je vidět příklad použití řetězení kódu s možností vícenásobného větvení kódu.

```
$('#content #canvas').loadApp(
  'data/delka_pexeso_meridla_pouziti.xml', function (data) {

    // po dohrání hry zapnout dialog výběru dalších
    aktivit

    var complete_dialog = data.find('complete_dialog');

    $('#content
    #canvas').addDialogWithHTMLHrefs(complete_dialog, '785px',
    '535px', 'slepa_mapa', function () {

      // vrátit zpět nápovědu

      jQuery.helps_data = data.find('data
      helps_data').clone();

      jQuery.help_show_id =
      jQuery.helps_data.find('default_help').text();

      jQuery.default_help_id = jQuery.help_show_id;

    });

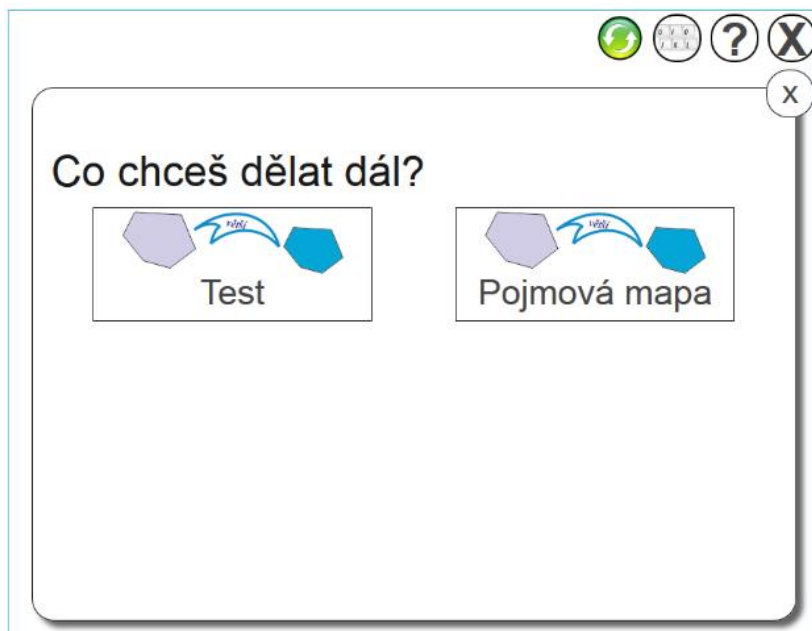
  });
```

Na tomto kódu je vidět, že po dohrání základní nahrané hry pexeso se provede načtení rozhodovacího dialogu, jenž je uložen v původním datovém souboru uvnitř elementu `complete_dialog`. Pro správné pochopení je potřeba ukázat ještě obsah tohoto datového souboru.

```
<complete_dialog>
  <helps_data>
    <default_help>vyber</default_help>
    <!-- Nápověda pro dialog -->
    <help id="vyber">
      <h2>Co chceš dál dělat?</h2>
    </help>
  </helps_data>
  <title>Co chceš dělat dál?</title>
  <data_link
    id="complete_test" class="dialog_click"
    data-targer_url="data/test.xml"
    data-background_image="images/bg-mapa-
    icon.png">Test</data_link>
  <data_link
    id="complete_pojmova_mapa"
    class="dialog_click"
    data-targer_url="data/posuvka.xml"
    data-background_image="images/bg-mapa-
    icon.png">Pojmová mapa</data_link>
</complete_dialog>
```

Výsledný dialog bude obsahovat 2 možné pokračování, jakým se může žák vydat.

Na následujícím obrázku je vidět výhodu takto univerzálně nastaveného návrhu načítání metod. Aniž by bylo potřeba načítat obsah celé webové stránky, můžou se do dialogů načítat klidně celé další úlohy.



OBR. 19 DIALOG ROZCESTNÍKU POKRAČOVÁNÍ



OBR. 18 UKÁZKA APLIKACE PUŠTĚNÉ V DIALOGU



## 5 Ověření navržených materiálů při výuce na ZŠ

Pro ověření vytvořených metod bylo zvoleno dvoustupňové ověření. První kontrolou bylo mé vlastní testování materiálů při výuce na hodinách průběžné i souvislé pedagogické praxe na ZŠ. Při těchto hodinách byla kontrolována jak technická, tak didaktická vhodnost výukových materiálů. Po odučení vedených hodin, byly diskutovány použité materiály s uvádějícím učitelem i učitelem dohlížejícím na průběžné pedagogické praxe a použité materiály byly přijaty velmi kladně. V této části ověření úloh již byly materiály v celkem stabilní fázi vývoje, a proto byl zjištěn pouze jeden technický problém s metodou křížovka. Jednalo se o drobný problém s vyplňováním křížovek na tabletech firmy Apple.

Děti byly použitím těchto netradičních materiálů motivovány a použití zlepšilo jejich pochopení probírané látky. Vyučující pan Mgr. Radovan Mikeš obzvláště ocenil využití pojmové mapy, jako průvodní mapy hodinou, kdy jednotlivé větve odpovídaly zadávaným úlohám či aktivitám. Díky názorné demonstraci měli žáci po celou dobu zajištěné pojmy a závislosti strukturované na interaktivní tabuli. Žáci, kteří odvodili fyzikální závislosti z daného pokusu, mohli za odměnu vyplnit odpovídající část pojmové mapy.

V druhé fázi evaluace byly poskytnuty předvyplněné materiály na téma „*délka*“ dvěma učitelům fyziky na základní škole. Jmenovitě se jednalo o doktorandy působící na katedře fyziky pány Víta Bednáře a Vladimíra Vochozku. Tito vyučující použili materiály při vlastních hodinách fyziky na druhém stupni ZŠ a postupně je takto testovali. V plném znění jsou zpětné reakce přiloženy v příloze této práce. Oba učitelé navržené materiály přijali kladně a ocenili převážně nezávislost materiálů na použitém hardware i platformě. Nejlépe hodnocené jsou pak motivační úlohy pexeso a křížovky spolu s opakovacími úlohami na procvičování převodů jednotek, které cituji „Výborné, nejvíc jsem využil, rozdělení podle obtížnosti je, jistě se z toho stane jedno z nejvyužívanějších podpor“ a úlohy testy. Během této fáze již nebyl zjištěn žádný nový technický problém s použitím materiálů.

## 6 Závěr

Při psaní této diplomové práce jsem si ověřil a zároveň prohloubil své znalosti technologie HTML5 a programování v jazyku Javascript. Čemu ovšem příkládám největší přínos této zkušenosti, tak to je samotná tvorba fyzikálních výukových materiálů pro základní školu. Nikdy bych si nedovedl představit, kolik práce je schováno za přípravou něčeho pro mě dříve tak samozřejmého, jako jsou školní učebnice.

V této práci se mi podařilo, splnit všechny cíle, které jsem si zadal. Vytvořil jsem pět funkčních výukových metod určených pro interaktivní tabule S.M.A.R.T Board a Activboard. Tyto metody jsem otestoval při výuce fyziky na 2. stupni ZŠ. Díky tomuto projektu jsem měl možnost vidět do zákulisí této tvorby. Poznal jsem při spolupráci lidi, kteří již něco ve svých oborech dokázali. Jako jsou páni autoři Jiří Tesař a František Jáchym, nebo pan ilustrátor Antonín Šplíchal, který ochotně realizoval mé představy v podobě krásných ilustrací použitých u rozcestníků jednotlivých úloh. Teprve až s použitím této profesionální grafiky získaly tyto materiály vzhled profesionality.

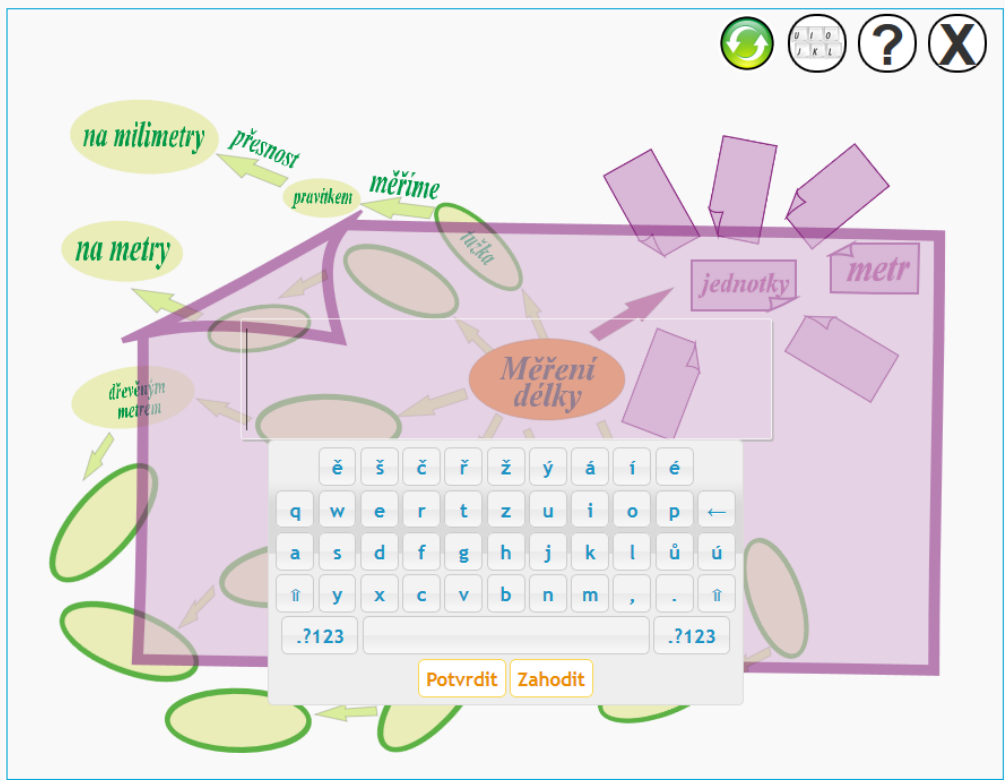
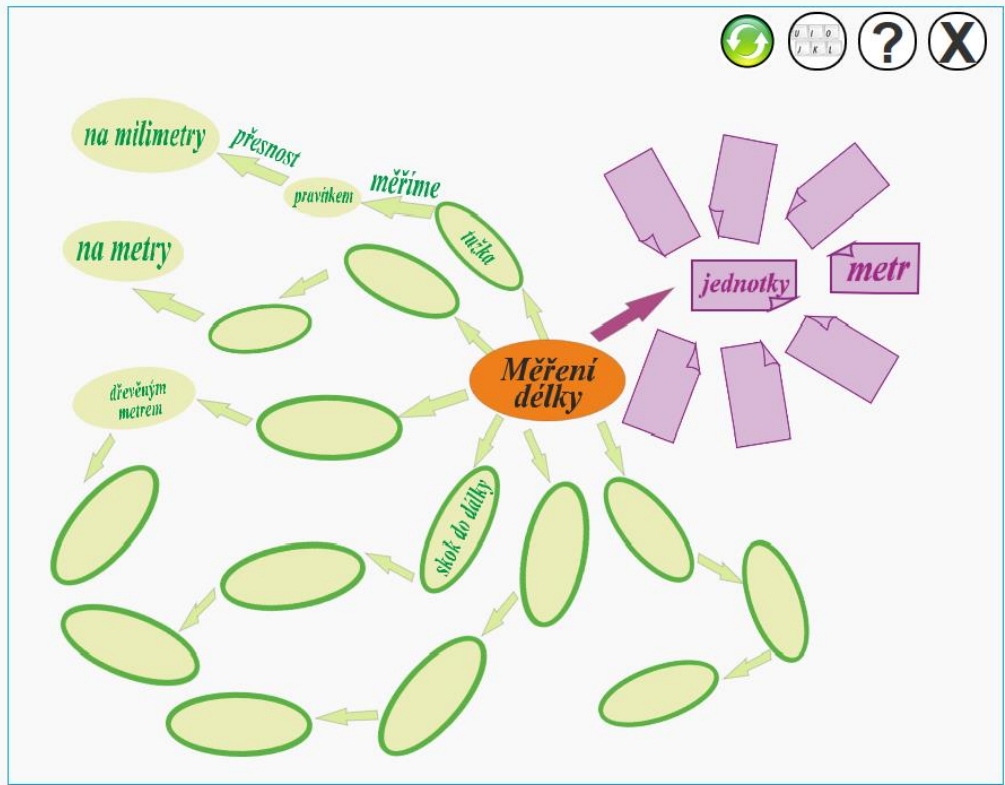
Obrovskou zkušeností pro mě byla komunikace s redaktory nakladatelství SPN. Při této práci jsem musel prostudovat velký počet materiálů a pracovat s více nástroji než za většinu doby během mého studia na vysoké škole. Velkým překvapením pro mě byl obrovský rozdíl v podpoře resp. nepodpoře současných technologií mezi současnými mobilními zařízeními. Bohužel stále platí pravda, že mnohem více času stráví programátor hledáním cest, jak donutit další zařízení či prohlížeč fungovat stejně, než programováním funkcionalit samotných.

Těší mě představa, že pravděpodobně budou mou práci používat učitelé při výuce fyziky a možná to třeba pomůže motivovat nějaké děti k dalšímu studiu fyziky. Odpracoval jsem na tomto projektu nespočet hodin tvrdé práce, která věřím, nebyla zbytečná.

## 7 Seznam použité literatury a zdrojů

- [1] EDUCATIONAL TECHNOLOGY. *The ADDIE Model: Instructional Design* [online]. 2014 [cit. 2015-03-08]. Dostupné z: <http://educationaltechnology.net/the-addie-model-instructional-design/>
- [2] W3.ORG. *Plan 2014* [online]. 2014 [cit. 2015-03-08]. Dostupné z: <http://dev.w3.org/html5/decision-policy/html5-2014-plan.html>
- [3] W3.ORG. Drag and Drop API [online]. 25.5.2011 [cit. 2012-03-25]. Dostupné z: <http://www.w3.org/TR/html5/dnd.html#dnd>
- [4] BARABÁŠ, Vít. *Drag and Drop API v HTML5*. České Budějovice, 2013. Bakalářská práce. Jihočeská univerzita. Vedoucí práce PaedDr. Petr Pexa, Ph.D.
- [5] Myšlenkové mapy. *KPI22 Kreativní práce s informacemi* [online]. 2011, č. 2 [cit. 2015-03-03]. Dostupné z: [https://www.mzk.cz/sites/mzk.cz/files/mentalni\\_mapy.pdf](https://www.mzk.cz/sites/mzk.cz/files/mentalni_mapy.pdf)
- [6] *Hodnocení citelnosti turistických map pro osoby s poruchou barvocitu*. Plzeň, 2006. Dostupné z: [http://gis.zcu.cz/studium/dp/2006/Spicelova\\_Hodnoceni\\_citelnosti\\_turistikych\\_map\\_pro\\_osoby\\_s\\_poruchou\\_barvocitu\\_BP.pdf](http://gis.zcu.cz/studium/dp/2006/Spicelova_Hodnoceni_citelnosti_turistikych_map_pro_osoby_s_poruchou_barvocitu_BP.pdf). Bakalářská. Západočeská univerzita. Vedoucí práce Ing. Mgr. Otakar Čerba.
- [7] JavaScript Array splice() Method. *W3schools.com* [online]. 2013 [cit. 2015-03-17]. Dostupné z: [http://www.w3schools.com/jsref/jsref\\_splice.asp](http://www.w3schools.com/jsref/jsref_splice.asp)





## Zpětná vazba od: Mgr. Vladimíra Vochozky

Vytvořené materiály korespondují se školním vzdělávacím plánem školy a stejně tak s rámcovým vzdělávacím plánem ministerstva školství pro základní vzdělávání. Jejich využití se nabízí k upevnění představy pojmu síla, které se vyskytuje v průniku látek od mechaniky po elektřinu s magnetismem.

Z hlediska softwarového a hardwarové vybavení školy (jiných škol) se jedná o aktuálně nejefektivnější řešení. Díky práci ve webovém prohlížeči odpadá nutnost zasahovat do nastavení zařízení, dodatečné instalace či využívání specifického hardwaru. Ovládání aktivity lze považovat za intuitivní a její běh v síti jako datově nenáročný.

Při doplňování pojmů lze předpokládat upevňování významu síla spojeného s třemi oblastmi výuky fyziky. Pro evokaci je použito zažitých příkladů spojených s typem síly (např. magnetické siločáry okolo pólů Země).

## Zpětná vazba od: Mgr. Vít Bednář

### Pojmové mapy

#### *Vyplněné mapy*

délka - ok

délka jiné - asi bych nedával vůbec, max. jako rozšiřující učivo, problém mají s "klasickými" jednotkami

délka 1 - ok

délka 2 - ok

#### *Neúplné mapy:*

jednotky délky - ok

jednotky délky - jiné - vynechat (zbytečně si s tím motají hlavu)

### Co je na obrázku

úloha 7 a 8 neznají

### Skládačky

skládačka 4 - neznají

### Pexeso - výborné

### Testy

ok - odpovídají náročnosti pro 6. ročník

### Doplňky informací

ok, ale přeřadil bych pod tuto kapitolu jednotky délky - jiné (ať jsou všechny doplňky u sebe)

### Převody jednotek

výborné, nejvíc jsem využil, rozdělení podle obtížnosti je, jistě se z toho stane jedno z nejvyužívanějších podpor

### Křížovky

dobré k domácímu procvičení, či jako DÚ

### Slovní a obrazové úlohy

velmi zdařilé, taktéž dobře použitelné do výuky, žáci zvládali, taktéž bych řekl  
stěžejní část

### Osobnosti

Rozšířit (pouze jedna) nebo bych spíš to naplnil zajímavým videem o té  
konkrétní osobě) zdroj youtube, různé pořady atd. )

Celkově zdařilý, určitě si v tom každý najde, co potřebuje do výuky doplnit -  
pro mě převody, slovní a obrazové úlohy, pexeso - nejčastěji použité