

UNIVERZITA PALACKÉHO V OLOMOUCI

PEDAGOGICKÁ FAKULTA

Katedra technické a informační výchovy

Bakalářská práce

Manipulace s objekty technikou „táhni a pusť“ v prostředí webu

Vojtěch Přecechtěl

Olomouc 2014

Vedoucí práce

Mgr. Jan Kubrický, Ph.D.

Prohlášení

Prohlašuji, že bakalářskou práci na téma Manipulace s objekty technikou „táhni a pusť“ v prostředí webu jsem vypracoval samostatně, uvedl a citoval všechny použité zdroje, které uvádím v jejich seznamu.

V Olomouci dne 21. 4. 2014

Podpis autora práce

Poděkování

Děkuji Mgr. Janu Kubrickému, Ph.D. za jeho cenné rady a čas při konzultacích a odborné vedení mé bakalářské práce. Také děkuji své rodině a všem, kdo se mnou měli v době psaní této práce trpělivost.

Obsah

ÚVOD	6
CÍL PRÁCE	7
TEORETICKÁ ČÁST	
1 TECHNIKA TÁHNI A PUSTĚ.....	9
1.1 VYMEZENÍ POJMU.....	9
1.2 HISTORIE	9
1.3 POUŽITÍ	10
1.4 DOSTUPNÉ METODY	11
1.4.1 HTML.....	11
1.4.2 JavaScript	11
1.4.3 Ajax	12
1.4.4 Flash.....	13
1.4.5 Silverlight	14
1.4.6 CSS.....	14
1.4.7 Java	15
1.4.8 Frameworky	15
2 HTML5.....	18
2.1 PŘEHLED NOVÝCH VLASTNOSTI	18
2.2 PODPORA WEBOVÝCH PROHLÍŽEČŮ	18
2.3 VALIDACE KÓDU	19
2.4 DRAG AND DROP.....	20
PRAKTICKÁ ČÁST	
3 DRAG AND DROP.....	25
3.1 UVEDENÍ DO PROBLEMATIKY	25
3.2 PLÁNOVÁNÍ A STANOVENÍ CÍLŮ	26

3.3	VÝVOJOVÁ PROSTŘEDÍ A NÁSTROJE.....	26
3.4	KÓD KOSTRY APLIKACE (HTML5)	27
3.5	KÓD VZHLEDU APLIKACE (CSS)	29
3.6	KÓD FUNKCÍ APLIKACE (JAVASCRIPT)	30
3.7	TEST APLIKACE	33
4	ZÁVĚR.....	34
	SEZNAM POUŽITÝCH ZDROJŮ LITERATURY.....	35
	CITACE TABULEK A OBRÁZKŮ	38
	SEZNAM PŘÍLOH.....	39
	PŘÍLOHA 1 - SEZNAM ZKRATEK A POJMŮ	40
	PŘÍLOHA 2 – SKRIPT PRO OVĚŘENÍ PODPORY HTML5 DND API	41
	PŘÍLOHA 3 – APLIKACE PRO NAČTENÍ ÚDAJŮ O PŘETAŽENÝCH SOUBORECH.....	42
	PŘÍLOHA 4 – HTML KÓD VÝUKOVÉ APLIKACE (APLIKACE.HTML).....	43
	PŘÍLOHA 5 – CSS KÓD VÝUKOVÉ APLIKACE (VZHLED.CSS).....	45
	PŘÍLOHA 6 – JAVASCRIPT KÓD VÝUKOVÉ APLIKACE (JAVASCRIPT.JS)	48
	ANOTACE.....	51

Poznámka

Některé termíny a zkratky nemusí být z textu srozumitelné, proto je uvádím vždy v poznámce pod čarou nebo v příloze č. 1, Seznamu zkratk a pojmů.

Úvod

S technikou „táhni a pusť“ se setkáváme v oblasti informačních technologií již dlouho a její implementace je dostupná snad ve všech současných počítačových operačních systémech s GUI¹. Používáme ji například ke kopírování souborů nebo posunování ikon nebo oken na ploše.

Tato technika se v poslední době začíná stále více používat i v UI² webových aplikací, které jsou budoucností, protože firmy vyvíjející software se orientují právě tímto směrem. Internet je samozřejmostí v každé domácnosti a přichází doba tzv. Cloud Computingu³ a webových aplikací.

V teoretické části se zaměřím na vymezení pojmu „táhni a pusť“, historii této techniky a uvedu podrobné informace a zajímavosti o této technice. Dále popíšu příklady použití, dostupné technologie, pomocí kterých lze této techniky u webových aplikací docílit. Název techniky „táhni a pusť“ je doslovný překladem z anglického „Drag and Drop“⁴, v obsahu práce budu používat anglický originál, protože v odborné literatuře se s ním setkáme častěji.

Velkou pozornost věnuji nové technologii HTML5, která je posledním standardem vývoje webových aplikací. Ověřím, zda techniku „táhni a pusť“ pomocí této technologie zvládnou implementovat i méně pokročilí uživatelé.

V praktické části naprogramuji a popíši vzorovou výukovou aplikaci, která by demonstrovala příklad použití techniky Drag and Drop zpracované pomocí HTML5.

Závěrem zhodnotím průběh, výsledky mé práce a také dosažení stanovených cílů.

¹ Graphical User Interface neboli grafické uživatelské rozhraní, je ovládané pomocí grafických prvků

² User Interface neboli uživatelské rozhraní

³ Poskytování služeb nebo programů uložených na serverech připojených k internetu

⁴ Gesto myši, které umožňuje táhnout objekt na obrazovce počítače, zkráceně DnD

Cíl práce

Cílem bakalářské práce je vytvořit přehled webových technologií, pomocí kterých lze manipulovat s objekty technikou „táhni a pusť“. Zaměřím se i na srovnání daných technologií z uživatelského i vývojářského hlediska.

Dalším cílem práce je využít získané teoretické znalosti a naprogramovat výukovou webovou aplikaci, na které demonstruji použití jazyka HTML5 a jeho DnD API⁵, případně doporučím jak postupovat, na co si dát pozor a čemu se vyhnout. Praktickým výstupem práce bude aplikace a její kód s podrobným komentářem a návodem.

⁵ HTML5 je značkový jazyk; Drag and Drop znamená „táhni a pusť“; Application Programming Interface je rozhraní pro programování aplikací, lze říci, že je to postup pro naprogramování určité funkce, v tomto případě techniky „táhni a pusť“.

TEORETICKÁ ČÁST

1 Technika táhni a pusť

V této kapitole představím funkci Drag and Drop, počátky jejího vzniku a použití, rovněž uvedu metody, které se používají pro naprogramování této dnes již oblíbené techniky.

1.1 Vymezení pojmu

Technika „táhni a pusť“ je uživatelská akce, kdy přesunujeme objekt pomocí myši po ploše obrazovky počítače, zatímco metoda je způsob, jak tuto uživatelskou akci naprogramovat. Přeložená definice termínu Drag and Drop z anglického výkladového slovníku do českého jazyka znamená: „posunout něco po počítačové obrazovce za použití myši a umístit to, kam chcete“. (1) Pro zařízení s dotykovou obrazovkou lze toto gesto také použít, místo držení levého tlačítka myši na objektu, který chceme přemístit, na objektu dlouze stiskneme displej, udržujeme styk s displejem a pohybem táhneme objekt, kam chceme a následným zvednutím prstu či stylusu ho umístíme. (2)

1.2 Historie

První osobní počítače se používaly v textovém prostředí a ovládaly pomocí příkazů psaných na klávesnici, jejich systémové prostředky byly omezené a neexistovalo uživatelské prostředí, kde by myš našla uplatnění. Počítačová myš byla sestrojena roku 1963, patentována 1970 a 1972 byla představena v podobě kuličkové myši, která poté byla uvedena na spotřebitelský trh, značně rozšířena a používána do nástupu myši optických a laserových. (3)

Roku 1973 Xerox představil svůj počítač Xerox Alto, který měl poprvé grafické uživatelské prostředí, které bylo možno ovládat myší. Prvním komerčním osobním počítačem, který se ovládal myší, byl Xerox 8010. Následovala ho společnost Apple Inc., která vyvinula vlastní uživatelské prostředí a nabídla ho roku 1983 komerčně ve svém počítači Apple Lisa. Obě sestavy byly natolik drahé, že je kupovaly většinou firmy. (4) Společnost Apple Inc. viděla nedostatky systému od Xeroxu, tak do svého systému vyvinula pro snazší ovládání techniku DnD, která se v té době nazývala Click and Drag. (2) Toto gesto se postupně stalo standardem pro GUI.

1.3 Použití

Název techniky Drag and Drop napovídá, že jde o přemísťování objektu na určité ploše, techniky lze tedy využít všude tam, kde toto uživatelské gesto myši lze aplikovat. Tato technika se často využívá jako alternativní a většinou existuje nejméně jedna další varianta, jak lze požadované akce dosáhnout pomocí jiného zařízení, například klávesnice.

V počátcích se technika DnD využívala například k uspořádání ikon nebo zobrazených oken na pracovní ploše a přesouvání souborů, postupně se tato metoda aplikovala všude tam, kde je to užitečné, dnes takto můžeme přiřadit soubor aplikaci, ve které ho chceme otevřít, přesunovat text, vkládat multimediální soubory do dokumentů, do webových služeb a aplikací. Většina současných operačních systémů a aplikací tuto techniku používá. (4)

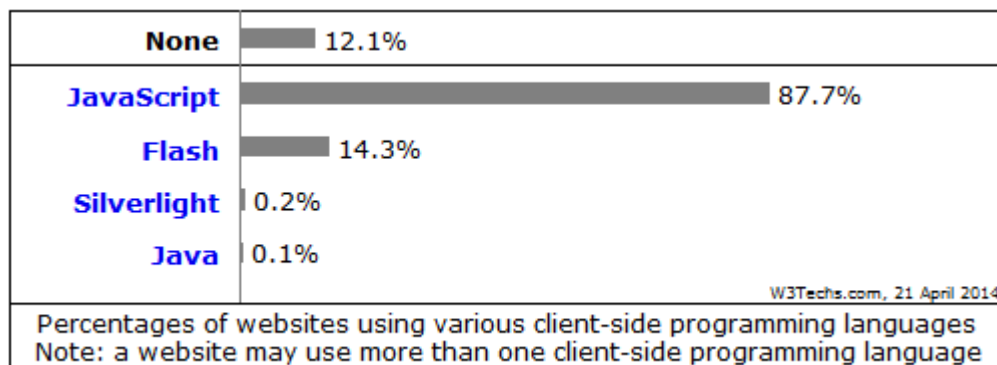
S růstem výpočetního výkonu, kapacity úložišť a hlavně rychlosti internetového připojení se internetový obsah začal vytvářet interaktivně, tak aby reagoval na uživatele a jejich požadavky, tak vznikly první webové aplikace. Přístup k internetovým službám se stal důležitým prvkem, pro spoustu lidí a firem nemá počítač bez přístupu k internetu vůbec smysl.

Webové prohlížeče jsou ve fázi, kdy zastávají některé funkce operačního systému, například zpřístupnění aplikací a služeb. Moderní webové aplikace se proto staly konkurenceschopnými těm desktopovým anebo jsou zcela originální a rozšiřují možnosti dále. Některé dokonce vypadají jako ty desktopové. Zejména cloudová úložiště a sociální sítě využívají techniky Drag and Drop ve svém uživatelském rozhraní pro umísťování multimediálních souborů.

Techniky Drag and Drop se využívá hlavně pro zvýšení uživatelského komfortu. Techniku lze vhodně zapojit do vyučovacího procesu, pokud máme ve třídě k dispozici interaktivní tabuli, počítače nebo tablety. Můžeme ji využívat například v aplikacích pro testování žáků, opakování látky nebo její názornou demonstraci. Je to jedna z mnoha cest, jak zaujmout žáky a udržet si jejich pozornost.

1.4 Dostupné metody

V této kapitole uvedu některé metody, které se využívají k implementaci techniky Drag and Drop pro objekty v prostředí webu, a krátce popíšu jejich vztah k technice DnD.



Graf 1- Procenta webových stránek užívající klientské programovací jazyky

1.4.1 HTML

HTML je zkratka HyperText Markup Language, jde tedy o hypertextový značkovací jazyk. Obsahuje ho každá webová stránka, je nejpoužívanějším a nejdůležitějším jazykem v prostředí webu, k jeho zápisu se používají tzv. tagy. Tim Berners-Lee navrhl první verzi HTML v roce 1990 spolu s klientským i serverovým softwarem, prvním webovým prohlížečem a protokolem HTTP. Později založil sdružení W3C, které pracuje na vývoji jazyka a jeho standardizaci. (6)

Pro techniku DnD je tento jazyk důležitý hlavně z hlediska informativního obsahu webu, který chceme přemístit, ať je to text, soubor nebo obrázky. V další kapitole se dočtete více informací o novém HTML5 DnD API, které je hlavní náplní této práce.

1.4.2 JavaScript

JavaScript je interpretovaným programovacím jazykem, to znamená, že program napsaný přímo do HTML kódu se odešle na klienta, a tam ho zpracuje a vyhodnotí internetový prohlížeč. (7, 8) Je to objektově orientovaný jazyk, využívá a pracuje s objekty prohlížeče, case sensitivní - v syntaxi záleží na velikosti písmen a svou syntaxí je podobný jazykům C nebo Java. Javascript může být uživatelem zakázán, nemusí být kompatibilní napříč verzemi prohlížečů ani jazyka samotného.

Omezení práce v tomto jazyku spočívá v tom, že data můžeme ukládat jen v rámci webu, v místě klienta pouze do cookies internetového prohlížeče, avšak nově lze také přistupovat do úložného prostoru webového prohlížeče. (7, 9)

Pomocí Javascriptu lze vyvolat interakci s uživatelem, která je důležitá i pro techniku DnD. Je důležitý pro zpracování informací u klienta, pro vyvolání uživatelských akcí se bez něj kód neobejde a v současné době není na webu nahraditelný, nebo je, ale jen částečně. (8) Podle statistik W3TECHS JavaScript využívá 87,9 % webových stránek. (Graf 1)

1.4.3 Ajax

„Technologie pro Ajax existovaly už v roce 1998 a také už je používaly některé aplikace, jako např. Microsoft Outlook Web Access. Ujaly se však až od začátku roku 2005, kdy je začaly využívat některé velmi používané aplikace, jako např. Google Suggest...“ (8)

Termín Ajax vznikl až roku 2005, kdy o něm napsal článek Jesse James Garret a vývojáři ho začali více používat. Ajax je ve skutečnosti kolekce technologií, slovo Ajax je zkratka pro „Asynchronous JavaScript and XML“.

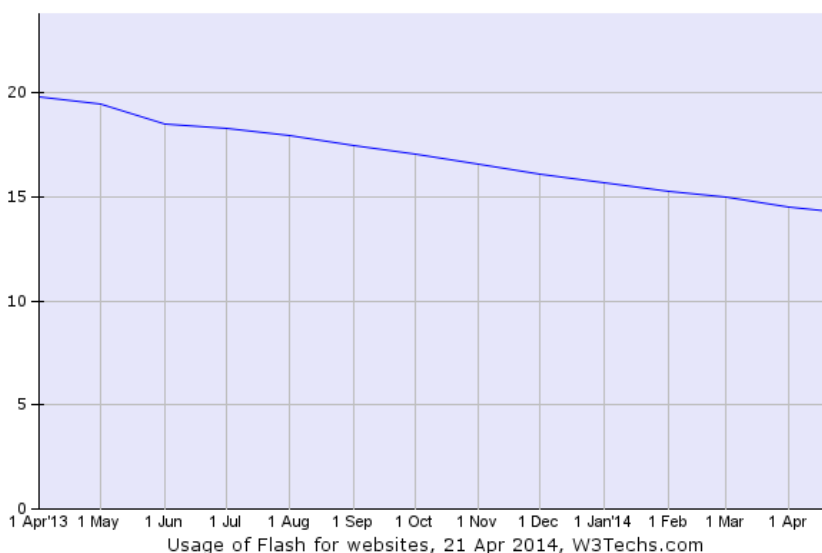
Termínem Asynchronous je myšleno to, že webový prohlížeč nečeká na odezvu serveru, aplikace běží stále, a data zpracuje poté, co je server odešle, ale uživatel nic nepozná, protože data se zpracují na pozadí. Jazyk JavaScript řídí operace, které prohlížeč vykonává. Zajišťuje na pozadí spojení se serverem, zpracování a zobrazení přijatých dat. (8, 10)

Jazyk XML je populární pro odesílání dat přes internet v textové podobě, je to jasně stanovený formát dat, proto se často používá a pomocí Javascriptu se data zpracují do formátu XML. Tato technika nemusí být použita, pokud chceme data zpracovávat jako čistý text. Kromě těchto technologií Ajax využívá dynamické HTML, kaskádové styly a PHP. (8) DnD v Ajaxu implementujeme pomocí Javascriptu, kdy použijeme události vyvolané myší nebo použijeme některou z mnoha knihoven. (11)

1.4.4 Flash

Flash je technologie firmy Adobe Systems, která umožňuje vytváření interaktivních aplikací. Uživatel musí mít nainstalovaný doplněk v prohlížeči, doplněk je dostupný pro operační systémy Windows, Mac a Linux. Pro mobilní zařízení byla ukončena podpora pro aplikaci Flash Player do webových prohlížečů. Flash pro tvorbu mobilních aplikací je nadále vyvíjen. Flash aplikace lze programovat v prostředí Adobe Flash Professional, které v současné verzi obsahuje i vývojové nástroje pro přímý export aplikace pro mobilní systémy iOS nebo Android. (12, 13, 14)

Flash pracuje s vektorovou i rastrovou grafikou a programuje se pomocí objektově orientovaného jazyku zvaného ActionScript. Tvorba aplikací není náročná, ale použití stále není univerzální. Pro uživatele webových Flash aplikací je nevýhodou nutnost instalovat doplněk do prohlížeče. Podle statistik W3TECHS se Flash používá na 14,3 % webových stránek (Graf 1). Při zhlédnutí statistik o využití technologie Flash při vývoji webových stránek společnosti W3Techs (Graf 2) můžeme vyčíslit, že v období duben 2013 až duben 2014 pokleslo využití technologie Flash o 5% webových stránek. Tento trend může zrcadlit nástup nových technologií.



Graf 2 - Webové stránky používající Flash (v procentech)

Příkladem použití Flash jsou například hry na serveru flashhry.cz, programy napsané v jazyku Flash se většinou vyznačují spuštěním v ohraničené oblasti na stránce, program však většinou lze maximalizovat.

Techniky Drag and Drop lze pomocí technologie Flash dosáhnout jen v rámci objektů dané Flash aplikace. Jazyk ActionScript je velice podobný JavaScriptu. Napsat Drag and Drop aplikaci v prostředí Flash je jednoduché a efektivní. Vývojářům firma Adobe nabízí ke koupi svůj software pro vývoj Flash aplikací, do kterého postupně implementuje prvky HTML5. (13) Manipulace s objekty technikou Drag and Drop pomocí Flash je velice podobná té, kterou si popíšeme v kapitole o HTML5. (15)

1.4.5 Silverlight

Silverlight je platforma určená pro tvorbu dynamického webového obsahu a interaktivní práce s ním. Kombinuje text, vektorovou i bitmapovou grafiku, animace a video. (16)

Stažení doplňku Silverlight do prohlížeče uživatelům umožní interaktivní ovládání her nebo aplikací a přehrávání multimédií ve většině současných webových prohlížečů (Internet Explorer, Firefox, Safari, Opera, Chrome) na platformách Windows a Mac OS X. (16) Pro GNU/Linux je dostupný pod názvem Moonlight, což je ukončený projekt společnosti Novell. (17)

Vývojové prostředí Silverlight je velice komplexní a výkonné, aplikace v něm napsané se mohou vyrovnat desktopovým, ale nevýhodou je nekompatibilita napříč operačními systémy, Silverlight nemůžeme použít na mobilních zařízeních, jako jsou mobilní telefony a tablety postavené na operačním systému Android. Vývoj Silverlight aplikací probíhá v uživatelském prostředí Microsoft Visual Studio nebo Microsoft Expression Studio. (16)

1.4.6 CSS

Cascading Style Sheets, česky kaskádové styly, jsou jazykem pro popis toho, jak se mají zobrazit webové stránky psané v HTML, XHTML nebo XML. První revize jazyka CSS byla napsána roku 1996, od té doby se podobně jako u HTML rozšiřovala syntax, která lze použít. V současné době je společně s HTML5 vyvíjena revize CSS3. (18)

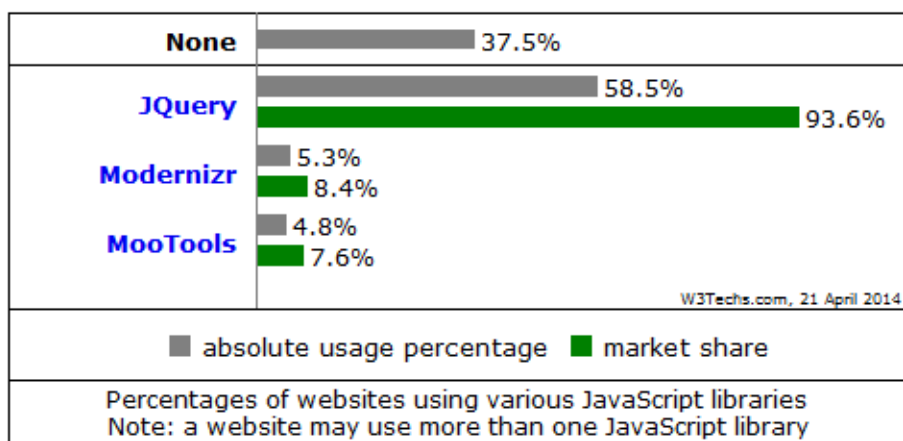
Kaskádovými styly ve spolupráci s technologiemi HTML5 a JavaScript lze nahradit technologii Flash. Mohou tak vznikat i složitější animace. (19, 20) Pro techniku Drag and Drop jsou důležité především pro správné zobrazení objektů, které jsou technikou DnD manipulovány. Pomocí kaskádových stylů definujeme vzhled objektů v rámci webového prohlížeče. (21) Zobrazení v každém prohlížeči se může lišit, proto je potřeba ověřit, jestli je aplikace použitelná v prohlížečích, které chceme podporovat.

1.4.7 Java

Java je objektově orientovaný programovací jazyk, který vyvíjí firma Sun Microsystems. Roku 1995 byla vydána první verze. Je jedním z nejpoužívanějších jazyků, protože lze spustit na snad každém zařízení. Roku 2007 Java začala být vyvíjena s otevřeným zdrojovým kódem. Uživatel musí mít nainstalovaný volně dostupný software, který umožní spuštění Java aplikací. Do doby, kdy internetové prohlížeče dostali podporu grafické hardwarové akcelerace pomocí WebGL, byl jazyk Java značně rychlejší a využitelnější pro graficky náročnější aplikace než například JavaScript, protože využíval hardwarovou akceleraci, avšak dnes je rychlost obou metod srovnatelná. (22) Webové aplikace psané v Javě se mohou otevírat v prohlížeči pomocí Java appletů (23) anebo lze otevřít i bez prohlížeče a to při použití frameworku Java Web Start. (24) Technika Drag and Drop se v Javě implementuje pomocí Java Drag and Drop API, které umožňuje přístup k událostem při tažení podobně jako HTML5 DnD API popsané v následující kapitole, implementace je o něco málo složitější.

1.4.8 Frameworky

Frameworky jsou hotová softwarová řešení dílčích problémů, které můžeme využít jako podporu při programování aplikací. Mohou obsahovat programy, návrhové vzory, doporučené postupy při vývoji anebo knihovny API (25, 26). Často se využívá i více frameworků v jednom projektu. V této části práce zmíním a popíšu ty více známé.



Graf 3 – Tři nejpoužívanější knihovny využívající JavaScript podle W3Techs

1.4.8.1 jQuery

Knihovna jQuery je psána v jazyce JavaScript, je rychlá, malá a obsahuje spoustu funkcí. Lze využít k manipulaci s dokumenty, správě událostí, animaci, a přístupu k Ajax API, které je kompatibilní s mnoha prohlížeči. (27) Podle serveru W3Techs jQuery využívá 58,5% webových stránek. (Graf 3) Takové využití přináší výhody, vývojáři mohou na internetu najít návody a čerpat nápady pro řešení svých problémů. Ti, kdo knihovnu používají, ví, že je k ní také k dispozici kvalitní dokumentace. Mimo jiné lze tuto knihovnu použít pro dosažení techniky táhni a pusť i bez implementace HTML5.

Pomocí jQuery můžeme nastavit objektům, které chceme manipulovat pomocí techniky DnD, parametr „draggable“ a pak s nimi manipulovat, zatímco cílovému elementu pro přesun nastavíme parametr „droppable“. (28) Výsledný kód je o něco delší, než když použijeme HTML5 DnD API. Řešení DnD pomocí jQuery nám přinese výhodu kompatibility i pro starší prohlížeče, které nepodporují HTML5. Můžeme využít širšího nastavení, lze například nastavit tolerance pro upuštění objektu pro dotykové obrazovky. (28, 29)

Zajímavý je i projekt s názvem jQuery Mobile, který se snaží o úplnou podporu všech zařízení. Využívá mimo jiné i HTML5 a s využitím tohoto API jednoduše dosáhneme responzivního web designu. (30)

SizzleJS je projekt od stejných tvůrců a zajišťuje výběr vhodného souboru CSS podle informací o zařízení, kde je aplikace spuštěna. (31)

1.4.8.2 MooTools

MooTools je knihovna, která využívá JavaScriptu a také umožňuje dosažení techniky DnD, je ale novější a není tak rozšířená. (Graf 3) Obsahuje velice přehlednou dokumentaci a je kompatibilní se všemi dnešními prohlížeči. Webové stránky s dokumentací obsahují příklady se zdrojovým kódem. Řešení s využitím této knihovny mi přišlo jednodušší a srozumitelnější, než HTML5 řešení. (32)

1.4.8.3 Modernizr

Modernizr je knihovna využitelná pro ověření podpory součástí HTML5 a CSS3 webového prohlížeče. Její implementací do hlavičky HTML kódu se rovněž umožní starším prohlížečům⁶ zobrazit některé nové HTML5 elementy a načíst CSS definice vzhledu. Knihovna nepřidává prohlížečům podporu CSS3, tagů audio nebo video, ale umožňuje zobrazení důležitých elementů stránky i ve starším webovém prohlížeči. (33)

⁶ Za starší webový prohlížeč lze považovat například Internet Explorer verze 6 až 9, který nemá podporu tagů HTML5.

2 HTML5

V této kapitole se podrobněji zaměříme na jazyk HTML5, který je současně asi nejdiskutovanější webovou technologií a poskytuje programátorům i uživatelům řadu výhod oproti ostatním řešením.

HTML5 je jazyk, který vychází ze svého předchůdce HTML4 a prakticky jen rozšiřuje jeho funkčnost. To znamená, že pokud jsme v minulosti napsali aplikaci v HTML4, bude nám v HTML5 fungovat a vypadat stále stejně, protože „tagy“ zůstaly stejné, jen některé přibyly. (34)

Použití jazyka HTML5 nám tedy přináší o něco více možností oproti předchozím verzím, se kterými můžeme pracovat. Můžeme hovořit o tom, že HTML5 je takové API prohlížeče, protože umožňuje přístup ke službám prohlížeče. (35)

2.1 Přehled nových vlastností

Nový standard HTML zahrnuje zjednodušení Doctype, Drag and Drop API, API pro geolokaci, spoustu nových tagů – úplnou novinkou jsou například párové tagy audio, video a canvas. (6)

2.2 Podpora webových prohlížečů

Podpora napříč webovými prohlížeči se zatím velmi různí, setkáme se tak s problémy při výběru formátů mediálních souborů. Zaleží na webovém prohlížeči, jaké formáty pro přehrávání audia a videa, API a funkce podporuje. Podpora kodeků napříč prohlížeči není taková, jakou by si vývojář představoval. Musí se ověřovat podpora prohlížeče pro použitý formát mediálních souborů, nastává nutnost uchovávat a poskytovat data ve dvou nebo více formátech a pomocí podmínek v JavaScriptu ošetřit načtení prohlížečem podporovaného souboru. Situace, kdy musíme soubor zakódovat pomocí dvou nebo tří kodeků, abychom ho mohli přehrát ve všech prohlížečích, jsou pravidlem.

To, zda webový prohlížeč podporuje DnD API můžeme ověřit pomocí Javascriptové knihovny Modernizr. Následující skript obsažený v souboru „testpodpory.html“ (Příloha 2) ověří podporu HTML5 DnD API v prohlížeči a vypíše hlášku s textem „HTML5 DnD API je podporováno.“, pokud jej podporuje. Při nekompatibilitě vypíše hlášku s textem: „Je potřeba mít JavaScript řešení.“. Stejným způsobem můžeme zvolit metodu, kterou má prohlížeč zpracovávat. Aby řešení bylo vždy kompatibilní, musíme aplikaci napsat jednou s využitím HTML5 a dále také alternativní variantu s využitím JavaScriptu a některé z dostupných knihoven.

```
<script src="modernizr.js">// knihovna pro ověření podpory HTML5</script>
<script>
if(Modernizr.dragandddrop) {
    // je možné řešení DnD pomocí html5
    alert("HTML5 DnD API je podporováno.");
} else {
    // je nutné nouzové řešení DnD pomocí JavaScriptu
    alert("Je potřeba mít JavaScript řešení.");
}
</script>
```

Zdrojový kód 1 – Skript pro ověření podpory prohlížeče pro HTML5 DnD API (Příloha 2)

2.3 Validace kódu

Webové stránky a aplikace by měly obsahovat kód HTML, který dodržuje normy. Vyhnete se tak nepříjemnostem, kdy zobrazení nemusí odpovídat kódu. Webové prohlížeče jsou psány a nastaveny pro uživatele, proto při výchozím nastavení nevypisují chyby v sintaxi, které stránky obsahují. Ale pokud bychom chtěli například exportovat stránky do formátu PDF, můžeme mít potíže. Také můžeme přijít na to, že každý prohlížeč zpracuje chybu jinak.

Automatickou kontrolu validity webové aplikace zvládnou pokročilejší vývojová prostředí, která jsou orientována primárně na vývoj webových aplikací. Validitu kódu si můžeme zdarma ověřit na stránkách s validátorem organizace W3C. (36)

2.4 Drag and Drop

HTML5 obsahuje drag and drop API, ke kterému můžeme jednoduše přistupovat pomocí JavaScriptu. U objektů a elementů HTML, které chceme manipulovat pomocí techniky DnD, nutně musíme nastavit atribut „draggable“ na hodnotu „true“. Dále je potřeba ovládat události, ve kterých budeme předávat data o pozici objektu a umožníme prohlížeči dokončit metodu DnD. (37, 38)

```

```

Zdrojový kód 2 – Příklad vložení obrázku kachna a nastavení atributu draggable

V následující tabulce jsou zobrazeny Drag and Drop události, které můžeme ovládat. Při těchto událostech můžeme z HTML dokumentu volat funkce psané v JavaScriptu. (38)

Událost	Popis průběhu události
dragStart	Spustí se, když uživatel začíná táhnout objekt
dragEnter	Spustí se, když poprvé táhneme objekt přes cílový element
dragOver	Je opakovaně vyvolána, dokud je tažený objekt nad elementem
dragLeave	Spustí se, když tažený objekt opustí plochu cílového elementu
drag	Spouští se při každém pohybu myši při přetahování objektu
drop	Spustí se při upuštění objektu do cílového elementu
dragEnd	Spustí se při ukončení tažení nebo upuštění objektu

Tabulka 1 - Popis průběhů událostí DnD API v HTML5

Pro dosažení funkční techniky DnD je nutné ošetření událostí „drag“, „dragOver“ a „drop“ funkcemi v JavaScriptu. Při události „drag“ načteme přenášená data do paměti. V události „dragOver“, která se spouští při tažení objektu nad cílovým elementem zajistíme možnost upuštění objektu tím, že zabráníme prohlížeči ve spouštění výchozí akce a při tažení objektu nad jiným objektem zakážeme upuštění. Nakonec ve funkci pro událost „drop“ načteme data z paměti a vložíme je do cílového elementu pomocí vlastnosti „appendChild“ cílového elementu „target“. V ukázce „ev“ znamená objekt pro danou událost, přidává se jako první objekt, když ošetřujeme událost.

```
ev.target.appendChild(document.getElementById(data));
```

Zdrojový kód 3 – Vložení obrázku do cílového elementu pomocí JavaScriptu

Ostatní události nemusíme využít, ale můžeme pomocí nich upravit proces DnD podle svých potřeb. Při události „dragStart“ nebo „drag“ můžeme nastavit obrázek při tažení. „dragEnter“ kombinaci s „dragLeave“ můžeme využít například pro změnu vzhledu cílových elementů. A v události „dragEnd“ nadefinujeme, co se má vykonat po upuštění objektu. (38)

Objekt „dataTransfer“ nám zajistí předávání dat objektů. Tomuto objektu můžeme podle potřeby definovat různé parametry. Nepovinná vlastnost „effectAllowed“ nám umožňuje nastavení povolené akce přenosu dat. Pokud vlastnost vůbec nenastavíme, tak jsou uživateli umožněny všechny akce. (39)

```
ev.dataTransfer.effectAllowed = 'hodnota';
```

Zdrojový kód 4 – Syntax nastavení vlastnosti objektu dataTransfer

Za hodnotu dosadíme hodnotu akce, kterou chceme povolit, možné hodnoty jsou uvedeny v tabulce 2. (39)

Hodnota	Popis účinku nastavené hodnoty
none	Zákaz přenosu dat, znemožní funkcionalitu DnD
copy	Tažený obsah smí být zkopírován
copyLink	Tažený obsah může být zkopírován nebo na něj odkazováno
copyMove	Tažený obsah může být zkopírován nebo přesunut
link	Na tažený obsah může být pouze odkazováno
linkMove	Tažený obsah může být přesunut nebo odkazován
move	Tažený obsah může být pouze přesunut
all	Tažený obsah může být zkopírován, přesunován i odkazován
uninitialized	Výchozí hodnota, stejný efekt jako hodnota „all“

Tabulka 2 - Hodnoty vlastnosti effectAllowed objektu dataTransfer a jejich účinky

Vlastností, která nám umožní zvolit jednu akci přenosu dat, je „dropEffect“. Pro tuto vlastnost lze nastavit hodnoty z tabulky 3. V zásadě vybíráme z hodnot povolených v hodnotě vlastnosti „effectAllowed“, tyto vlastnosti mají mnohdy stejný výsledný efekt. Tyto vlastnosti se nastavují při události „dragEnter“ nebo „dragOver“. (39)

Hodnota	Popis účinku nastavené hodnoty
copy	Vytvoří se kopie zdrojového objektu v cílovém umístění
move	Zdrojový objekt se přesune na cílové umístění
link	V cílovém umístění se vytvoří odkaz na zdrojový objekt
none	Objekt nebude možné upustit

Tabulka 3 - Hodnoty vlastnosti dropEffect objektu dataTransfer a jejich účinky

Pro uložení dat o taženém objektu do paměti používáme vlastnost „setData“. V syntaxi této vlastnosti nastavujeme hodnoty 2 parametrů.

```
ev.dataTransfer.setData(format, data);
```

Zdrojový kód 4 – Příklad načtení dat o přetahovaném objektu do paměti

U parametru „format“, nastavujeme formát přenášených dat. Jeho hodnotu lze nastavit například na hodnotu 'text' pro text nebo 'text/uri-list' pro odkaz. Nastavit je možné jakýkoliv jiný datový MIME typ. Zdá se, že při nastavení formátu dat například na hodnotu 'image/jpeg' vznikají komplikace s kompatibilitou webových prohlížečů (např. u Internet Explorer), proto jsem používal hodnotu pro text i pro přenos odkazů obrázků. (40, 41)

Do parametru „data“ zadáme data, která chceme přenášet, většinou odkazujeme na přenášený objekt pomocí cesty k elementu nebo hypertextovým odkazem. Pokud již existují a jsou načtena v paměti data stejného formátu, budou přepsána.

Data budeme chtít poté použít a zpracovat, musíme je tedy načíst z paměti pomocí vlastnosti „getData“, v hodnotě se blíže specifikuje jen formát dat.

```
ev.dataTransfer.getData(format);
```

Zdrojový kód 5 – Příklad získání dat o přetahovaném objektu z paměti

Pomocí vlastnosti „clearData“ můžeme data po přenesení a upuštění objektu z paměti smazat. Není to nezbytné, ale je to korektní. Pro tuto vlastnost se nastavuje jen formát dat, který se má odstranit. Buď se implementuje do události „dragEnd“ nebo na konec události „drop“.

Další vlastností objektu dataTransfer je „setDragImage“, pomocí které můžeme nastavit obrázek, který se má pod kurzorem myši zobrazovat při tažení objektu. U této vlastnosti se jako hodnota blíže specifikuje daný element, použijeme k tomu odkaz na soubor s obrázkem anebo element. Dále nastavíme souřadnice polohy kurzoru myši vůči obrázku v pořadí x, y. Při použití následujícího zápisu se bude kurzor myši při přetahování v levém horním rohu obrázku.

```
ev.dataTransfer.setDragImage('obrazek.jpg', 0, 0);
```

Zdrojový kód 6 – Příklad nastavení obrázku pod kurzorem myši při tažení objektu

Pokud chceme použít techniku DnD pro manipulaci se soubory, chceme například přetáhnout soubor z uživatelské plochy do prohlížeče, je to možné pomocí vlastnosti objektu „dataTransfer.files“.

Vše, co jsme si řekli, platí i zde, jen vynecháme ošetření událostí „drag“ a „dragStart“, nutné je tedy jen ošetření událostí „dragOver“, kdy musíme povolit upuštění a „drop“, kdy předáme získaná data do některého elementu prohlížeče. Kód jednoduché aplikace pro načtení údajů o přetažených souborech, který jsem našel u společnosti Dottoro, okomentoval ho a přeložil, najdete v souboru „soubory.html“ (42, Příloha 3). Kód je dostupný k edukačním účelům. (42)

PRAKTICKÁ ČÁST

3 Drag and Drop

3.1 Uvedení do problematiky

Nyní si představíme, jak postupovat při tvorbě HTML5 aplikace, která využívá techniku Drag and Drop. Připravil jsem jednoduchou příkladovou aplikaci, na které budu vše demonstrovat. Tato aplikace je vlastně učební pomůckou, která lze ovládat pomocí myši či interaktivní tabule a má za úkol rozšířit nebo ověřit slovní zásobu anglického jazyka žáků. Jde o přiřazování obrázků k anglickým slovíčkům. Finální verzi aplikace vidíte na obrázku.



Obrázek 1 - Drag and Drop výuková aplikace

3.2 Plánování a stanovení cílů

Každý vývojář musí mít představu, jak má výsledná aplikace vypadat a jaké budou její funkce. Také je potřeba získat povědomí o dostupných prostředcích. Podle toho se volí prostředky k dosažení cílů. V mém případě jsem zvolil kombinaci HTML5, JavaScriptu a CSS3. Pomocí těchto prostředků je možné vyvinout výukovou aplikaci, ve které by uživatel přiřazoval obrázky k textu, a zároveň bylo by možné kontrolovat správnost přiřazení a vypsát hodnocení úspěšnosti.

Při volbě prostředků je nutné myslet na to, kde budeme aplikaci využívat a na jakých zařízeních budeme aplikaci spouštět. Velkou roli hraje například velikost a rozlišení obrazovky. Pro tuto aplikaci jsem stanovil minimální možné rozlišení obrazovky 1024 x 768 pixelů, kdy půjde aplikaci pohodlně ovládat. Toto rozlišení jsem si ospravedlnil statistikami W3Schools, ze kterých vyplývá, že většina uživatelů dnes již používá vyšší rozlišení. (43)

Abychom aplikaci vytvářeli efektivně, je dobré dělat si v průběhu plánování poznámky, aplikaci si načrtnout, popsat si funkce, které se budeme snažit implementovat. To vše nám práci ulehčí a zajistí, že kód aplikace bude mít určitou strukturu, které se budeme moci držet.

3.3 Vývojová prostředí a nástroje

Podle použitých prostředků si můžeme vybrat vývojové prostředí. Pro jednoduché úpravy stačí jakýkoliv textový editor, lepší je ale použít rozumné prostředí, které zvýrazňuje a nabízí syntax kódu. Zaujalo mě komplexní prostředí s přehledným nastavením a zobrazením Aptana Studio 3, podobného zobrazení jsem dosáhl s prostředím Komodo Edit, NetBeans a Eclipse, velice zajímavý byl jednoduchý editor Sublime Text 2, který umožňoval vše, co jsem potřeboval a jeho nároky na systém byly v porovnání se zmíněnými prostředími minimální. Za vyzkoušení stojí i sada softwaru Adobe Edge, která umožňuje i tvorbu animací v HTML5 a CSS3 (20) nebo známý Adobe Dreamweaver.

3.4 Kód kostry aplikace (HTML5)

Nejdříve si vytvoříme kostru aplikace. Potřebujeme objekty, které později budeme ovládat a nastavovat pomocí CSS a JavaScriptu. Vytvoříme soubor, který nazveme například „aplikace.html“ (Příloha 4) Na začátku kódu si můžete všimnout zjednodušení „doctype“. Také vidíme standardní strukturu HTML dokumentu.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Drag and drop - Bakalářská práce</title>
  <link rel="stylesheet" type="text/css" href="vzhled.css">
  <script src="javascript.js"></script>
</head>
<body>
</body>
</html>
```

Zdrojový kód 7 – Ukázka struktury HTML dokumentu, nastavení a načtení prvků v hlavičce

V hlavičce definujeme mimo titulku, který se zobrazuje v popisu okna prohlížeče i znakovou sadu, kterou jsem zvolil UTF-8 a také načítáme externí soubory, které si předem vytvoříme. V tomto případě je to soubor „vzhled.css“, který obsahuje údaje o vzhledu prvků v dokumentu a soubor „javascript.js“, který bude obsahovat kód JavaScript. Je dobré kód oddělit od struktury dokumentu a od jeho vzhledu.

Do těla dokumentu vložíme objekty tedy obrázky⁷, které budeme chtít manipulovat technikou Drag and Drop. Vložil jsem je do elementů DIV s ID „drag1“ až „drag10“, aby bylo možné jim nastavit parametry pozadí a okrajů. U obrázků je důležitý atribut slovíčku. Pro dosažení techniky Drag and Drop je nutné u obrázků nastavit atribut „id“, který později využijeme pro vyhodnocení, jestli je obrázek přiřazen ke správnému „draggable“ na hodnotu „true“.

⁷ Pro potřeby své práce jsem použil vektorové obrázky ze sad Simple Farm Animals 1 a 2 (46), které jsou volně šiřitelné, tímto bych chtěl poděkovat jejich autorovi, uživateli webu Deviantart.com s přezdívkou Viscious-Speed.

```
<div id="drag1">

</div>
```

Zdrojový kód 8 – Ukázka vložení obrázku kachna s nastavenými atributy a událostmi

Potom přidáme události „ondragstart“ a „ondragover“, které má prohlížeč sledovat a které budou odkazovat na kód v JavaScriptu a volat funkce „drag“ a „zakaz“, které si popíšeme později. Událostí můžeme sledovat a využít i více, jsou popsány v teoretické části práce.

Také potřebujeme mít prostor, kam obrázky budeme moci přetahovat. Vytvořil jsem element DIV s ID „slovicka“ a do něj umístil elementy DIV s ID „cil1“ až „cil10“, které budou obsahovat anglická slovíčka a budou se k nim přiřazovat obrázky. Je potřeba sledovat události „ondrop“ a „ondragover“, kterými voláme funkce „drop“ a „allowDrop“, které si napíšeme v kódu JavaScriptu.

```
<div id="slovicka">
<div id="cil1" ondrop="drop(event)" ondragover="allowDrop(event)">Dog</div>
</div>
```

Zdrojový kód 9 – Ukázka cílového elementu s nastavenými událostmi

Do aplikace jsem ze zajímavosti přidal ještě sledování událostí, kterými ovládám vzhled cílových elementů se slovíčky při tažení obrázku na element a pryč, ve funkci „vyplnit“ odstraním pozadí a funkcí „vyplnpryc“ pozadí zase doplním.

```
ondragenter="vyplnit(event)" ondragleave="vyplnpryc(event)"
```

Zdrojový kód 10 – Nastavení událostí pro ovládání nastavení vzhledu cílových elementů

Do těla dokumentu jsem na závěr přidal dvě tlačítka: „tlacitko1“, které slouží k vyvolání kontroly správnosti přiřazení obrázků ke slovíčkům, po kliknutí se zavolá funkce „kontrola“. A „tlacitko2“, které restartuje aplikaci.

```
<div id="tlacitka">
<button type="button" id="tlacitko1" onClick="kontrola(event)">
Check!
</button>
<button type="button" id="tlacitko2" onClick="location.reload(true)">
Start Again
</button>
</div>
```

Zdrojový kód 11 – Ukázka přidání tlačítek do těla dokumentu a jejich atributů

3.5 Kód vzhledu aplikace (CSS)

CSS3 se stará o vzhled aplikace, pro navržení jednoduchých vzhledů doporučuji přehledný online nástroj na webu CSS Portal pro tvorbu rozložení prvků, tzv. layoutů, který za nás nadefinuje strukturu CSS šablony, užitečné jsou i další nástroje, využil jsem například generátor vzhledu tlačítek. (44)

Pro vzhled aplikace je potřeba nadefinovat všechny prvky tak, abychom dosáhli požadovaných efektů. Technika Drag and Drop bude samozřejmě fungovat i bez definování vzhledu, v příloze 5 je možné zhlédnout celý CSS kód souboru „vzhled.css“.

Pro správnou implementaci Drag and Drop se po najetí ukazatelem myši nad objekt, který má atribut „draggable“, nastaví ukazatel myši tak, aby nám ukazoval možnost tažení objektu. Přidal jsem tedy k obrázkům řádek s příkazem pro nastavení kurzoru.

```
cursor: move;
```

Zdrojový kód 12 – Definice stylu kurzoru při jeho poloze nad obrázkem

Při řešení vzhledu cílových elementů jsem narazil na problém. Pomocí CSS pseudotřídy „hover“ nebylo možné docílit změny vlastností elementu při tažení objektu nad element. Podobné řešení se mi podařilo aplikovat při najetí ukazatelem myši na výchozí elementy „drag1“ až „drag10“, ve kterých jsou umístěny obrázky. U cílových elementů jsem docílil požadovaného efektu, kdy po tažení obrázku nad cílový element zmizí otazník v pozadí elementu, pomocí ošetření událostí „dragEnter“ a „dragLeave“ funkcemi v JavaScriptu.

Dále pro dosažení techniky Drag and Drop nepotřebujeme v CSS definovat nic zvláštního, postupujeme stejně jako při definování kaskádových stylů běžných webových stránek.

3.6 Kód funkcí aplikace (JavaScript)

Soubor „javascript.js“ (Příloha 6) musí obsahovat všechny funkce, které voláme z dokumentu „aplikace.html“ (Příloha 4). Všechny uvedené funkce kromě funkce „kontrola“ jsou nezbytné pro zajištění správné funkčnosti této metody.

Funkce „drag“ je volána při události „ondragstart“, funkcí chceme povolit přesun obrázku, proto nastavíme hodnotu „ev.dataTransfer.effectAllowed“ na „move“, na začátku události drag načteme data o taženém obrázku do paměti pomocí „dataTransfer.setData“, kdy v závorce definujeme formát dat, u obrázků je dobré používat formát „text“, vyhneme se problémům s kompatibilitou napříč prohlížeči pro formát „image“. Pokud chceme nastavit obrázek, který se má zobrazovat při tažení, můžeme do funkce přidat řádek „ev.dataTransfer.setDragImage (odkaz na obrázek, poloha x, poloha y);“

```
/* funkce, která se spustí a trvá při události tažení obrázku, načítáme údaje o tažené položce */  
function drag(ev)  
{  
    ev.dataTransfer.setData('text',ev.target.id);  
}
```

Zdrojový kód 13 – Funkce pro načtení dat o obrázku do paměti při události drag

Chceme, aby bylo možné obrázek upustit do vytvořených elementů DIV s ID „cil1“ až „cil10“. Při události „ondragover“ proto voláme funkci „allowDrop“, ve které řekneme prohlížeči, že nechceme provádět akci, která je pro něho výchozí. A také můžeme nastavit požadovaný efekt přenosu dat pomocí vlastností objektu „dataTransfer“.

```
/* povolení upuštění objektu, zakazuje výchozí událost */  
function allowDrop(ev)  
{  
    ev.preventDefault();  
    ev.dataTransfer.effectAllowed = 'move';  
    ev.dataTransfer.dropEffect = 'move';  
}
```

Zdrojový kód 13 - Funkce pro povolení upuštění obrázku při události dragOver

Dalším krokem je pro nás funkce „drop“, která má za úkol načtení dat do příslušného cílového elementu DIV při upuštění obrázku. V souboru „aplikace.html“ na ni odkazujeme pomocí události „ondrop“. Funkce musí zakázat výchozí událost prohlížeče, jinak by nás přesměroval na přetažený obrázek. Dále načteme do proměnné „data“ ID obrázku, které jsme již uložili do paměti ve funkci „drag“. Přiřadíme obrázek do cílového elementu DIV. Aplikujeme styly pro změnu vzhledu na objekty a ukončíme práci s objektem.

```
/* funkce, která se spustí při upuštění objektu */
function drop(ev) {
/* bez tohoto zákazu výchozí události se po akci drop přesměruje stránka na
tažený obrázek */
ev.preventDefault();
/* načteme data do proměnné */
var data = ev.dataTransfer.getData('text');
/* načtená data vložíme do cílového elementu */
ev.target.appendChild(document.getElementById(data));
/* při upuštění dojde ke změně CSS hranic obrázků a výmazu pozadí */
ev.target.style.border = '5px solid';
ev.target.style.backgroundImage='none';
document.getElementById(data).style.backgroundImage = 'none';
/* drop je poslední akcí s objektem, tímto příkazem s objektem přestaneme
pracovat */
ev.stopPropagation();
ev.dataTransfer.clearData('text'); } /* vymažeme přenášená data */
```

Zdrojový kód 13 – Funkce ošetření upuštění objektu

Zdá se, že by to mohlo být vše, protože „drop“ je poslední funkce, kterou musíme zavolat, aby přetahování obrázků bylo funkční. Avšak když si aplikaci v této fázi vyzkoušíme, tak se zjistíme, že pokud přetáhneme obrázek na cílový element DIV, který již obsahuje jiný obrázek, tak aktuálně přetahovaný obrázek zmizí. Musíme ošetřit, aby se tak nestalo, proto zakážeme upuštění obrázku nad jiným obrázkem nebo cílovým elementem DIV. Využijeme události „ondragover“, ve které zavoláme funkci „zakaz“.

```
/* zákázání události drop - zákaz upuštění obrázku na jiný obrázek */
function zakaz(ev) {
/* pro Google Chrome a Mozilla Firefox dostačuje */
ev.dataTransfer.effectAllowed = 'none';
ev.dataTransfer.dropEffect = 'none';
/* musí se přidat kvůli IE, zdá se, že ani verze 11 nepodporuje vlastnost
effectAllowed */
ev.preventDefault = 'false'; }
```

Zdrojový kód 14 – Funkce pro zákaz upuštění obrázku nad jiným obrázkem

Funkce „zakaz“ zablokuje přenos dat díky nastavení vlastnosti „dataTransfer.effectAllowed“ na hodnotu „none“. V prohlížeči Internet Explorer 11 tento parametr neměl žádný efekt, proto jsem nechal povolené výchozí akce prohlížeče, které taktéž znemožní upuštění objektu.

Většinou nestačí objekt někam jen přesunout, chceme ho přesunout, abychom mohli získat informace o jeho nové pozici a data zpracovat. Proto vytvoříme novou funkci s názvem „kontrola“. Tato funkce zajišťuje v našem případě kontrolu správnosti přiřazení obrázků zvířátek k anglickým slovíčkům. Musíme tedy vědět, které zvířátko jsme kam přiřadili. Každý obrázek má jedinečné ID shodné s anglickým slovíčkem pro vyobrazené zvířátko, které se porovná s textovým obsahem každého cílového elementu DIV, který obsahuje anglické slovíčko pro zvířátko.

Využil jsem cyklu typu „for“, pomocí kterého provedu v dokumentu výběr všech obrázků s atributem „draggable“. Cyklem porovnáím postupně ID obrázků s textem v cílových elementech DIV a na základě toho provedu obarvení políček s obrázky. Pokud je přiřazení správné, tak je okraj políčka a text zelený, jinak se změní na červenou. Také počítám správné odpovědi, ze kterých určím úspěšnost z poměru správného a celkového počtu přiřazení.

V této aplikaci je nutné dbát na nastavení ID obrázků, které se musí shodovat s některým slovíčkem v cílovém elementu DIV. Na základě této skutečnosti probíhá ověření správnosti přiřazení. Tento způsob ověřování jsem vybral kvůli jednoduchosti, s jakou lze dosáhnout kontroly přiřazení. Zároveň jsem po zvážení ponechal názvy obrázků české, aby při uložení obrázku nebylo možné zjistit jeho anglický překlad.

Pro žáky je vhodné, když hned nevidí správné výsledky a musí více přemýšlet, proto správnost přiřazení kontroluji až tehdy, když uživatel uzná za vhodné a klikne na tlačítko s textem „Check!“, které zavolá funkci „kontrola“. Kompletní komentovaný kód funkce je dostupný v přílohách. (Příloha 6)

3.7 Test aplikace

Aplikaci jsem spustil v 5 nejznámějších webových prohlížečích na platformě Windows. Vytvořil jsem tabulku, která shrnuje získané poznatky.

Prohlížeč	Komentář
Mozilla Firefox 28	Vše funkční.
Google Chrome 34	Vše funkční.
Opera 20	Vše funkční.
Internet Explorer 11	Nezná některé CSS3 styly, podporuje jen některé objekty jazyka JavaScript a typy souborů, technika DnD funkční.
Safari 5.1.7	Prohlížeč neumožní upuštění objektu na cílový element. Zobrazení aplikace je v pořádku.

Tabulka 4 - Test aplikace v prohlížečích na platformě Microsoft Windows

V prohlížečích, které jsou označeny zeleně, byla aplikace korektně zobrazena a technika DnD byla funkční. Internet Explorer 11 zatím plně nepodporuje práci s CSS3 a vektorovými obrázky, proto jejich zobrazení nebylo korektní, ale DnD technika byla funkční. Naopak tomu bylo u Safari 5.1.7, který poskytl správné zobrazení všech prvků aplikace, ale přetažení obrázku nebylo možné dokončit.⁸

Zajímavé je, že všechny z testovaných prohlížečů (i Safari) při použití skriptu pro ověření podpory prohlížeče (Příloha 2) hlásily, že HTML5 DnD API podporují.

Ani po letech, kdy je koncept HTML5 dostupný, s ním není dostupný software plně kompatibilní. Uvidíme, jestli se situace změní, až organizace W3C uvolní koncem roku 2014 stabilní verzi HTML 5.0 (6). Pro psaní DnD webové aplikace je tedy zatím lepší použít některou z dostupných knihoven, která zajistí kompatibilitu a širší možnosti nastavení.

⁸Toto je chyba Safari verze 5.1.7 pro platformu MS Windows, na platformě Mac je technika DnD funkční.

4 Závěr

V teoretické části práce jsem vymezil pojem Drag and Drop, uvedl jsem jeho historii, pokusil jsem se popsat současný vývoj webových technologií a také rozvedl využití techniky DnD. Na konci první kapitoly jsem vytvořil přehled metod, které bychom mohli použít pro naprogramování funkce DnD pro prostředí webu.

Ve druhé kapitole jsem se zabýval problematikou HTML5, podporou webových prohlížečů a validací kódu. Pro ověření podpory HTML5 DnD API jsem vytvořil a okomentoval jednoduchý skript využívající knihovnu Modernizr, který je součástí příloh. Detailně jsem popsal nové HTML5 DnD API a možnosti, které nabízí. V rámci teoretické části práce odkazuji na jednoduchou aplikaci, kterou jsem přeložil a důkladně okomentoval, protože mi posloužila jako učební pomůcka při získávání znalostí o DnD manipulaci se soubory.

Praktickou část jsem pojal jako průvodce tvorbou výukové Drag and Drop aplikace, proto jsem okomentoval a popsal funkce jednotlivých částí kódu, které jsem rozdělil podle využívané technologie. V přílohách k práci najdeme části kódu jednotlivých souborů popsaných v práci. Zároveň tato aplikace slouží k demonstraci využití manipulace s objekty technikou Drag and Drop. V přílohách uvádím i další, pomocné aplikace, které jsem využil při psaní práce.

Vytvořil jsem přehled dostupných technologií pro danou problematiku a k vybrané metodě jsem shromáždil dostatek informací pro vývoj aplikace. Tvorba výukové aplikace pro demonstraci DnD pomocí nového jazyka HTML5 mě mnohé naučila a aplikace je funkční tak, jak jsem zamýšlel.

HTML5 je mladá technologie, přesto se s ní pracovalo dobře a těším se na její rozšíření. Při vývoji komplikovanější webové aplikace bych zatím raději volil implementaci DnD pomocí některé z dostupných knihoven, které mají ošetřeny problémy s kompatibilitou prohlížečů a obsahují některé pokročilejší funkce, které vývojářům usnadňují vývoj. Výslednou aplikaci plánuji v budoucnosti rozšířit o další funkce a GUI pro administraci, kde by bylo možné měnit parametry aplikace a do aplikace vkládat nové objekty.

Seznam použitých zdrojů literatury

- (1) Macmillan Dictionary. *Drag and Drop* [online]. [cit. 2013-3-11] Dostupné z WWW: <http://www.macmillandictionary.com/dictionary/british/drag-and-drop>
- (2) *Drag and Drop* [online]. 18. 4. 2014 [cit. 2014-18-4] Dostupné z WWW: http://en.wikipedia.org/wiki/Drag_and_drop
- (3) *Mouse (computing)* [online]. [cit. 2013-3-11] Dostupné z WWW: http://en.wikipedia.org/wiki/Mouse_%28computing%29
- (4) *History of the graphical user interface* [online]. 10. 4. 2014 [cit. 2014-11-4] Dostupné z WWW: http://en.wikipedia.org/wiki/History_of_the_graphical_user_interface
- (5) *The Lisa* [online]. [cit. 2013-5-11] Dostupné z WWW: <http://www.vectronicsappleworld.com/macintosh/lisagui.html>
- (6) *HTML* [online]. 11. 4. 2014 [cit. 2014-13-4] Dostupné z WWW: <http://en.wikipedia.org/wiki/HTML>
- (7) *JavaScript* [online]. 13. 4. 2014 [cit. 2014-14-4] Dostupné z WWW: <http://en.wikipedia.org/wiki/JavaScript>
- (8) RESIG, J. *Javascript a Ajax: Moderní programování webových aplikací*. Brno, Computer Press, a. s., 2007. ISBN: 978-80-251-1824-5
- (9) *JavaScript - Úvod* [online]. [cit. 2014-14-4] Dostupné z WWW: <http://www.jakpsatweb.cz/javascript/javascript-uvod.html>
- (10) HOLZNER, S. *Mistrovství v AJAXu*. Vyd. 1., Brno: Computer Press a. s., 2007. 591 stran. ISBN 978-80-251-1850-4. (str. 21 – 23)
- (11) Tometa Software, Inc. *Ajax Tutorial: Drag & Drop* [online]. 2006 [cit. 2014-15-4] Dostupné z WWW: <http://www.tometasoftware.com/Ajax-Tutorial-Drag-Drop.asp>
- (12) Adobe Flash Catalyst CS5: oficiální výukový kurz. Vyd. 1. Brno: Computer Press a. s., 2011. 280 s. ISBN 978-80-251-3254-8.
- (13) *Flash* [online]. 2006 [cit. 2014-15-4] Dostupné z WWW: <http://www.adobe.com/cz/products/flash.html>
- (14) *Adobe Flash* [online]. 11. 4. 2014 [cit. 2014-15-4] Dostupné z WWW: http://en.wikipedia.org/wiki/Adobe_Flash
- (15) *Flash drag and drop tutorial* [online]. 28. 1. 2012 [cit. 2014-16-4] Dostupné z WWW: http://edutechwiki.unige.ch/en/Flash_drag_and_drop_tutorial
- (16) *Silverlight* [online]. [cit. 2013-5-11] Dostupné z WWW: <http://msdn.microsoft.com/en-us/silverlight/default>
- (17) *Moonlight* [online]. 6. 2. 2014 [cit. 2014-5-3] Dostupné z WWW: http://en.wikipedia.org/wiki/Moonlight_%28runtime%29
- (18) *Kaskádové styly* [online]. 27. 3. 2014 [cit. 2014-5-4] Dostupné z WWW: http://cs.wikipedia.org/wiki/Kask%C3%A1dov%C3%A9_styly
- (19) CALZADILLA, A. *Pure CSS3 Spiderman Cartoon w/ jQuery and HTML5* [online]. 3. 4. 2010 [cit. 2014-5-2] Dostupné z WWW: <http://www.optimum7.com/internet-marketing/web-development/pure-css3-spiderman-ipad-cartoon-jquery-html5-no-flash.html>

- (20) *Adobe Edge Animate CC* [online]. [cit. 2014-5-4] Dostupné z WWW:
<http://html.adobe.com/edge/animate/>
- (21) *CSS Tutorial* [online]. [cit. 2014-10-4] Dostupné z WWW:
<http://www.w3schools.com/css/>
- (22) *Java (programovací jazyk)* [online]. 13. 4. 2014 [cit. 2014-15-4] Dostupné z WWW:
[http://cs.wikipedia.org/wiki/Java_\(programovací_jazyk\)](http://cs.wikipedia.org/wiki/Java_(programovací_jazyk))
- (23) *Java applet* [online]. 13. 4. 2014 [cit. 2014-15-4] Dostupné z WWW:
http://en.wikipedia.org/wiki/Java_applet
- (24) *Java Web Start* [online]. 9. 4. 2014 [cit. 2014-15-4] Dostupné z WWW:
http://en.wikipedia.org/wiki/Java_Web_Start
- (25) *Framework* [online]. 11. 2. 2014 [cit. 2014-10-3] Dostupné z WWW:
<http://cs.wikipedia.org/wiki/Framework>
- (26) *API* [online]. 12. 4. 2014 [cit. 2014-15-4] Dostupné z WWW:
<http://cs.wikipedia.org/wiki/API>
- (27) *jQuery* [online]. 2014 [cit. 2014-5-4] Dostupné z WWW: <http://jquery.com/>
- (28) *Draggable* [online]. [cit. 2014-2-4] Dostupné z WWW:
<http://api.jqueryui.com/draggable/>
- (29) *jQuery drag and drop example* [online]. 2. 6. 2013 [cit. 2014-5-4] Dostupné z WWW: <http://www.overpie.com/jquery/articles/jquery-drag-and-drop-example>
- (30) *jQuery Mobile* [online]. 2014 [cit. 2014-4-4] Dostupné z WWW:
<http://jquerymobile.com>
- (31) *SizzleJS* [online]. 2014 [cit. 2014-4-4] Dostupné z WWW:
<http://sizzlejs.com/>
- (32) MooTools. *Drag.Drop* [online]. 2014 [cit. 2014-5-4] Dostupné z WWW:
<http://mootools.net/demos/?demo=Drag.Drop>
- (33) *Modernizr* [online]. 2014 [cit. 2014-6-4] Dostupné z WWW: <http://modernizr.com/>
- (34) ALBERS, B. et al. *HTML5: programujeme moderní webové aplikace*. Vyd. 1. Brno: Computer Press a. s., 2011. 304 s. ISBN 978-80-251-3539-6.
- (35) CASTRO, E. & HYSLOP, B. *HTML5 a CSS3: národní průvodce tvorbou WWW stránek*. 1. vyd. Brno: Computer Press a. s., 2012. 439 s. ISBN 978-80-251-3733-8.
- (36) *W3C Markup Validation Service* [online]. 2014 [cit. 2014-6-4] Dostupné z WWW:
<http://validator.w3.org/>
- (37) Mozilla Foundation. *Drag Operations* [online]. 2014 [cit. 2014-6-3] Dostupné z WWW: https://developer.mozilla.org/en-US/docs/DragDrop/Drag_Operations
- (38) Mozilla Foundation. *Drag and Drop* [online]. 2014 [cit. 2014-6-3] Dostupné z WWW: https://developer.mozilla.org/en-US/docs/DragDrop/Drag_and_Drop
- (39) Mozilla Foundation. *DataTransfer* [online]. 2013 [cit. 2014-6-3] Dostupné z WWW: <https://developer.mozilla.org/en-US/docs/Web/API/DataTransfer>
- (40) Mozilla Foundation. *Recommended Drag Types* [online]. 2011 [cit. 2014-6-3] Dostupné z WWW: https://developer.mozilla.org/en-US/docs/DragDrop/Recommended_Drag_Types
- (41) *MIME* [online]. 2014 [cit. 2014-8-4] Dostupné z WWW:
<http://en.wikipedia.org/wiki/MIME>
- (42) *files property (dataTransfer)* [online]. [cit. 2014-10-4] Dostupné z WWW:
<http://help.dottoro.com/ljslrhdh.php>

- (43) *Screen Resolution Statistics* [online]. [cit. 2014-15-4] Dostupné z WWW:
http://www.w3schools.com/browsers/browsers_display.asp
- (44) *Layout Generator*. [online]. [cit. 2014-15-4] Dostupné z WWW:
<http://www.cssportal.com/layout-generator/>
- (45) *Cloud computing* [online]. [cit. 2014-20-4] Dostupné z WWW:
http://cs.wikipedia.org/wiki/Cloud_computing
- (46) *Simple Farm Animals 1, 2*. [online]. 2012. [cit. 2014-20-4] Dostupné z WWW:
<http://openclipart.org/search/?query=simple+farm+animals>
- (47) ANDERSON, S. P. *Přitažlivý interaktivní design: jak vytvářet uživatelsky přívětivé produkty*. 1. vyd. Brno: Computer Press, 2012. 240 s. ISBN 978-80-251-3722-2.
- (48) DARIE, C. et al. *AJAX a PHP: tvoříme interaktivní webové aplikace profesionálně*. Vyd. 1. Brno: Zoner Press, 2006. 320 s. ISBN 80-86815-47-1.
- (49) LACKO, L. *Ajax: Hotová řešení*. 1. vydání. Brno: Computer Press a. s., 2008. 269 s. ISBN 978-80-251-2108-5

Citace grafů, obrázků a tabulek

Graf 1 - W3Techs. *Procenta webových stránek užívající klientské programovací jazyky.*

21. 4. 2014. Dostupný z WWW:

http://w3techs.com/technologies/overview/client_side_language/all

Graf 2 - W3Techs. *Webové stránky používající Flash (v procentech).* 21. 4. 2014.

Dostupný z WWW: <http://w3techs.com/technologies/details/cp-flash/all/all>

Graf 3 - W3Techs. *Tři nejpoužívanější knihovny využívající JavaScript podle W3Techs.*

21. 4. 2014. Dostupný z WWW:

http://w3techs.com/technologies/overview/javascript_library/all

Obrázek 1 - *Drag and Drop výuková aplikace.* Obrázek je snímkem obrazovky počítače autora práce, který je autorem vyobrazené aplikace.

Tabulka 1 – *Popis průběhů událostí DnD API v HTML5.* Vytvořena podle: *Drag and Drop*

[online]. 2014 [cit. 2014-6-3] Dostupné z WWW: [https://developer.mozilla.org/en-](https://developer.mozilla.org/en-US/docs/DragDrop/Drag_and_Drop)

[US/docs/DragDrop/Drag_and_Drop](https://developer.mozilla.org/en-US/docs/DragDrop/Drag_and_Drop)

Tabulka 2 – *Hodnoty vlastnosti effectAllowed objektu dataTransfer a jejich účinky.*

Vytvořena podle: *DataTransfer* [online]. 2013 [cit. 2014-6-3] Dostupné z WWW:

<https://developer.mozilla.org/en-US/docs/Web/API/DataTransfer>

Tabulka3 – *Hodnoty vlastnosti dropEffect objektu dataTransfer a jejich účinky.*

Vytvořena podle: *DataTransfer* [online]. 2013 [cit. 2014-6-3] Dostupné z WWW:

<https://developer.mozilla.org/en-US/docs/Web/API/DataTransfer>

Seznam příloh

Příloha 1 – Seznam zkratk a pojmů

Příloha 2 – Skript pro ověření podpory prohlížeče HTML5 DnD API (testpodpory.html)

Příloha 3 – Aplikace pro načtení údajů o přetažených souborech (soubory.html)

Příloha 4 – HTML kód výukové aplikace (aplikace.html)

Příloha 5 – CSS kód výukové aplikace (vzhled.css)

Příloha 6 – JavaScript kód výukové aplikace (javascript.js)

Poznámka

Autor práce je autorem všech uvedených příloh kromě přílohy číslo 3, kde byl kód aplikace autorem práce pouze přeložen do češtiny a okomentován.

Příloha 1 - Seznam zkratk a pojmů

Cloud Computing

Cloud computing je poskytování služeb nebo programů uložených na serverech připojených k internetu s tím, že uživatelé k nim mohou přistupovat a využívat jejich služeb například pomocí webového prohlížeče nebo klientské aplikace. (45)

Drag and Drop (DnD)

Táhni a pusť je gesto prováděné počítačovou myší, gesto se skládá ze čtyř fází, nejprve přesuneme ukazatel myši nad objekt, stiskneme a držíme levé tlačítko myši, přesuneme objekt, kam chceme a nakonec uvolníme tlačítko myši, abychom objekt umístili. (1, 2)

API

API je zkratka pro Application Programming Interface, v českém překladu „rozhraní pro programování aplikací“. API se používají i pro přístup a předávání informací. Webové služby jako Facebook nebo Dropbox poskytují vývojářům API pro propojení aplikací s jejich službami. Tyto API umožňují vznik aplikací pro mobilní a jiná zařízení – například i ty, které nejsou provozovatelem služby podporovány. Za použití API můžeme rozdělit programovaný kód a spojit více programovacích jazyků pro dosažení cíle. (26)

Framework

je hotové softwarové řešení dílčích problémů, které můžeme využít jako podporu při programování aplikací. Mohou obsahovat programy, návrhové vzory, doporučené postupy při vývoji anebo knihovny API (25, 26). Často se využívá i více frameworků v jednom projektu.

PHP

Hypertextový preprocesor je serverový skriptovací programovací jazyk využívaný pro programování dynamických webových stránek a aplikací.

GUI

Grafické uživatelské rozhraní je rozhraní ovládané pomocí grafických prvků, jako jsou okna oken, panely, tlačítka a další objekty. (4)

UI

User Interface neboli uživatelské rozhraní.

Příloha 2 – Skript pro ověření podpory HTML5 DnD API

Obsah souboru „testpodpory.html“:

```
<!DOCTYPE html>
<html>
<title>Test podpory HTML5 DnD API</title>
<head>
<script src="modernizr.js">
// knihovna pro ověření podpory HTML5</script>
<script>
if(Modernizr.draganddrop) {
    // je možné řešení dnd pomocí html5
    alert("HTML5 DnD API je podporováno.");
} else {
    // je nutné nouzové řešení dnd pomocí javascriptu
    alert("Je potřeba mít JavaScript řešení.");
}
</script>
</head>
</html>
```

Příloha 3 – Aplikace pro načtení údajů o přetažených souborech

Obsah souboru „soubory.html“:

```
<!DOCTYPE html>
<html lang="cs">
<head>
  <meta charset="UTF-8">
  <title>HTML5 File Drag and Drop API</title>
  <script>
/* fce pro zákaz upuštění souboru na tělo dokumentu a vypsání chybové
hlášky */
  function OnDropBody (event) {
    alert ("Prosím, přetáhněte své soubory do textového pole.");
    return false;
  }
/* fce ošetření upuštění objektu */
  function drop (event) {
    /* podporuje prohlížeč objekt dataTransfer? */
    if (event.dataTransfer) {
/* podporuje prohlížeč vlastnost files objektu dataTransfer? */
      if (event.dataTransfer.files) {
/* pokud ano, načteme cílový element do proměnné target */
        var target = document.getElementById ("target");
/* vyprázdníme textbox */
        target.value = "";
/* pomocí cyklu for načteme postupně všechny upuštěné soubory */
        for (var i = 0; i < event.dataTransfer.files.length; i++)
/* stanovení proměnných a načtení údajů o souboru */
          { var file = event.dataTransfer.files[i];
            if ('name' in file) {var fileName = file.name;}
            else { var fileName = file.fileName; }
            if ('size' in file) { var fileSize = file.size; }
            else { var fileSize = file.fileSize; }
/* nakonec vypíšeme proměnné v řetězci do cílového textového pole */
            target.value += (i+1) + ". Soubor -- název: " +
            fileName + ", velikost: " + fileSize + " bytů\n"; } }
        else { alert ("Váš prohlížeč nepodporuje vlastnost pro soubory."); }}

    else { alert ("Váš prohlížeč nepodporuje vlastnost objektu
dataTransfer.");} if (event.stopPropagation) {event.stopPropagation ();}
    else {event.cancelBubble = true;}
    return false;}
  </script>
</head>

<body ondragenter="return false;" ondragover="return false;"
ondrop="return OnDropBody (event);">
  Do následujícího pole můžete přetáhnout soubor/y:<br />
  <textarea id="target" rows="5" cols="50" spellcheck="false"
    ondragenter="return false;" ondragover="return false;"
    ondrop="return drop (event);">
  </textarea>
</body>
</html>
```

Příloha 4 – HTML kód výukové aplikace (aplikace.html)

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Drag and drop - Bakalářská práce</title>
  <link rel="stylesheet" type="text/css" href="vzhled.css">
  <script src="javascript.js"></script>
</head>

<body>
<div id="ram">
<div id="hlavicka">
<h1>Match following pictures with English words:</h1>
</div>

<div id="ohraniceniobrazku">

<div id="drag1"></div>

<div id="drag2"></div>

<div id="drag3"></div>

<div id="drag4"></div>

<div id="drag5"></div>

<div id="drag6"></div>

<div id="drag7"></div>

<div id="drag8"></div>

<div id="drag9"></div>

<div id="drag10"></div>

</div>
```

```

<div id="kontejnernaobrazky">
  <div id="slovicka">

<div id="cil1" ondrop="drop(event)" ondragover="allowDrop(event)"
ondragenter="vyplnit(event)" ondragleave="vyplnpryc(event)">Dog
</div>

<div id="cil2" ondrop="drop(event)" ondragover="allowDrop(event)"
ondragenter="vyplnit(event)" ondragleave="vyplnpryc(event)">Duck
</div>

<div id="cil3" ondrop="drop(event)" ondragover="allowDrop(event)"
ondragenter="vyplnit(event)" ondragleave="vyplnpryc(event)">Hen
</div>

<div id="cil4" ondrop="drop(event)" ondragover="allowDrop(event)"
ondragenter="vyplnit(event)" ondragleave="vyplnpryc(event)">Cow
</div>

<div id="cil5" ondrop="drop(event)" ondragover="allowDrop(event)"
ondragenter="vyplnit(event)" ondragleave="vyplnpryc(event)">Horse
</div>

<div id="cil6" ondrop="drop(event)" ondragover="allowDrop(event)"
ondragenter="vyplnit(event)" ondragleave="vyplnpryc(event)">Rabbit
</div>

<div id="cil7" ondrop="drop(event)" ondragover="allowDrop(event)"
ondragenter="vyplnit(event)" ondragleave="vyplnpryc(event)">Cat
</div>

<div id="cil8" ondrop="drop(event)" ondragover="allowDrop(event)"
ondragenter="vyplnit(event)" ondragleave="vyplnpryc(event)">Pig
</div>

<div id="cil9" ondrop="drop(event)" ondragover="allowDrop(event)"
ondragenter="vyplnit(event)" ondragleave="vyplnpryc(event)">Sheep
</div>

<div id="cil10" ondrop="drop(event)" ondragover="allowDrop(event)"
ondragenter="vyplnit(event)" ondragleave="vyplnpryc(event)">Suricata
</div>

</div>
</div>
<div id="tlacitka">
<button type="button" id="tlacitko1" onclick="kontrola(event)">Check!
</button>

<button type="button" id="tlacitko2" onclick="location.reload(true)">
Start Again
</button>

</div>
</div>
</body>
</html>

```

Příloha 5 – CSS kód výukové aplikace (vzhled.css)

```
@import url("reset.css");

body {
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 13px;
}

h1 {
    font-size: 24px;
    font-family: arial;
    font-weight: bold;
    text-shadow: 2px 2px 5px #EA0000;
    color: #9A0000;
    padding: 10px;
}

p {
    padding: 10px;
}

#ram {
    margin: 0 auto;
    width: 860px;
}

#hlavicka {
    width: 800px;
    height: 40px;
    float: left;
    border-radius: 10px;
    border: 2px solid #eb9524;
    margin: 4px;
    background: #63B8EE;
    background: linear-gradient(top, #EE5700, #FFCF4A);
    background: -ms-linear-gradient(top, #EE5700, #FFCF4A);
    background: -webkit-gradient(linear, left top, left bottom,
from (#EE5700), to (#FFCF4A));
    background: -moz-linear-gradient(top, #EE5700, #FFCF4A);
}

#ohraniceniobrazku {
    max-width: 800px;
    max-height: 500px;
    float: left;
    background: #FFFFFF;
    border-radius: 10px;
    border: 5px dotted #FFCC73;
    margin: 4px;
}
}
```

```

#drag, #drag1, #drag2, #drag3, #drag4, #drag5, #drag6, #drag7, #drag8,
#drag9, #drag10 {
    background: #FFFFFF;
    border-radius: 10px;
    border: 5px solid #FFCC73;
    margin: 5px;
    width: 140px;
    height: 140px;
    float: left;
}

#drag:hover, #drag1:hover, #drag2:hover, #drag3:hover, #drag4:hover,
#drag5:hover, #drag6:hover, #drag7:hover, #drag8:hover, #drag9:hover,
#drag10:hover {
    border: 5px dotted #FFCC73;
}

#drag img, #drag1 img, #drag2 img, #drag3 img, #drag4 img, #drag5 img,
#drag6 img, #drag7 img, #drag8 img, #drag9 img, #drag10 img {
    margin: 5px;
    width: 130px;
    height: 130px;
    bottom: 0px;
    cursor: move;
}

#kontejnernaobrazky {
    width: 810px;
    max-height: 500px;
    float: left;
    background: #FFFFFF;
    margin: 4px;
}

#slovicka {
    background: #FFAF38;
    border-radius: 10px;
    margin: 5px;
}

#cil1, #cil2, #cil3, #cil4, #cil5, #cil6, #cil7, #cil8, #cil9, #cil10 {
    background: #FFFFFF;
    border-radius: 10px;
    border: 5px dashed #FFCC73;
    margin: 5px;
    width: 140px;
    height: 160px;
    float: left;
    font-size: 2em;
    text-align: center;
    text-shadow: -1px 0 black, 0 1px black, 1px 0 black, 0 -1px black;
    color: #FFA938;
    background-image: url("../obrazky/question.svg");
    background-repeat: no-repeat;
}

```

```

#cil1 img, #cil2 img, #cil3 img, #cil4 img, #cil5 img, #cil6 img, #cil7
img, #cil8 img, #cil9 img, #cil10 img {
    margin: 5px;
    width: 130px;
    height:130px;
    bottom: 0px;
    cursor: move;
}

#tlacitka {
    width: 300px;
    height: 10px;
    padding: 10px;
}

#tlacitko1, #tlacitko2 {
    font-family: arial;
    font-weight: bold;
    color: #6A0000 !important;
    font-size: 14px;
    text-shadow: 1px 1px 5px #DE0000;
    box-shadow: 1px 1px 1px #F9C578;
    padding: 6px 25px;
    -moz-border-radius: 10px;
    -webkit-border-radius: 10px;
    border-radius: 10px;
    border: 2px solid #A30000;
    background: #63B8EE;
    background: linear-gradient(top, #EE5700, #FFCF4A);
    background: -ms-linear-gradient(top, #EE5700, #FFCF4A);
    background: -webkit-gradient(linear, left top, left bottom,
from(#EE5700), to(#FFCF4A));
    background: -moz-linear-gradient(top, #EE5700, #FFCF4A);
}

#tlacitko1:hover, #tlacitko2:hover {
    color: #FFB70F !important;
    background: #468ccf;
    background: linear-gradient(top, #B30000, #EE1A1A);
    background: -ms-linear-gradient(top, #B30000, #EE1A1A);
    background: -webkit-gradient(linear, left top, left bottom,
from(#B30000), to(#EE1A1A));
    background: -moz-linear-gradient(top, #B30000, #EE1A1A);
}

```

Příloha 6 – JavaScript kód výukové aplikace (javascript.js)

```
//nahrajeme data o obrázku do paměti
function drag(ev)
{
ev.dataTransfer.setData('text',ev.target.id);
}

/* povolení upuštění objektu, zakazuje výchozí událost */
function allowDrop(ev)
{
ev.preventDefault();
ev.dataTransfer.effectAllowed = 'move';
ev.dataTransfer.dropEffect = 'move';
}

/* po najetí taženým objektem nad cílový element zmizí otazník z pozadí*/
function vyplnit(ev)
{
ev.target.style.backgroundImage='none';
document.getElementById(data).style.backgroundImage = 'none';
}

/* po najetí taženým objektem pryč z cílového elementu se otazník v
pozadí zase objeví */
function vyplnpryc(ev)
{
ev.target.style.backgroundImage='url(/obrazky/question.svg)';
document.getElementById(data).style.backgroundImage='url(/obrazky/questi
on.svg)';
}

/* funkce, která se spustí po upuštění objektu */
function drop(ev)
{
/* bez tohoto zákazu výchozí události se po akci drop přesměruje stránka
na tažený obrázek */
ev.preventDefault();
/* načteme data do proměnné */
var data = ev.dataTransfer.getData('text');
/* načtená data vložíme do cílového elementu */
ev.target.appendChild(document.getElementById(data));
/* při upuštění dojde ke změně CSS hranic obrázků a výmazu pozadí */
ev.target.style.border = '5px solid';
ev.target.style.backgroundImage='none';
document.getElementById(data).style.backgroundImage = 'none';
/* drop je poslední akcí s objektem, tímto příkazem s objektem přestaneme
pracovat */
ev.stopPropagation();
/* vymažene přenášená data */
ev.dataTransfer.clearData('text');
}
```



```

/* zákazání události drop - zákaz upuštění obrázku na jiný obrázek */
function zakaz(ev)
{
ev.dataTransfer.effectAllowed = 'none'; /* pro Google Chrome a Mozilla
Firefox dostačuje */
ev.dataTransfer.dropEffect = 'none'; /* pro Google Chrome a Mozilla
Firefox dostačuje */
/* musí se přidat kvůli IE, zdá se, že ani verze 11 nepodporuje
vlastnosti effectAllowed a effectDrop */
ev.preventDefault = 'false';
}

/* kontrola přiřazení obrázků ke slovíčkům pomocí cyklu for a vyhodnocení
dosaženého skóre*/
function kontrola(ev)
{
/* stanovení proměnných pro cyklus */
/* načteme všechny objekty s tagem img */
var images = document.getElementsByTagName('img');
var i;
var imgId;
var divId;
var divText;

/* stanovení proměnných pro počítání výsledků a hodnocení */
var celkem = 10;
var dobre = 0;
var procenta;

/* cyklus, vyhodnocujeme polohu obrázku a porovnááme s textem obsaženým
v cílovém objektu div*/
for (i = 0; i < images.length; i++)
/* jen pokud obrázek obsahuje atribut draggable, tak s ním dále pracujeme
*/
{ if (images[i].getAttribute('draggable'))
{
/* načtení získaných hodnot do proměnných */
imgId = images[i].id;
divText = images[i].parentNode.textContent;
divId = images[i].parentNode.id;
var cestakdivu = document.getElementById(divId).style;
/* porovnááme text v elementu s id obrázku, následuje vyhodnocení
proměnných pomocí podmínky*/
if (divText == imgId)
/* pokud je obrázek přiřazen správně, otazník v pozadí zmizí, změní se
barva textu a okraj objektu div na zelenou */
{
cestakdivu.backgroundImage='none';
cestakdivu.color = 'green';
cestakdivu.border='5px solid green';
/* přičítáme číslo 1, pokud je obrázek přiřazen správně */
dobre = dobre+1;
}
}
}

```

```

else
/* pokud je obrázek přiřazen špatně, otazník v pozadí zmizí, změní se
barva textu a okraj objektu div na červenou */
{
cestakdivu.backgroundImage='none';
cestakdivu.color = 'red' ;
cestakdivu.border='5px solid red';
}
}
}
/* Vyhodnocení výsledků */
/* stanovíme podíl správného počtu odpovědí a celkového počtu a násobíme
100, abychom dostali procenta úspěšnosti */
procenta = dobre/celkem*100;
/* pokud je úspěšnost vyšší, než 20%, tak se zobrazí skore, ale aplikace
vám již nedovolí se opravit, skryjeme tlačítko1 */
if (procenta>20)
{
alert('Vaše skore je '+ procenta + ' %');
document.getElementById('tlacitko1').style.visibility='hidden';
}
else
{
alert('Máte velice nízké skore ' + procenta + ' %, můžete si ho vylepšit
a poté znovu zkontrolovat. ');
}
}

```

Anotace

Jméno a příjmení:	Vojtěch Přecechtěl
Katedra:	Katedra technické a informační výchovy
Vedoucí práce:	Mgr. Jan Kubrický, Ph.D.
Rok obhajoby:	2014
Název práce:	Manipulace s objekty technikou „táhni a pusť“ v prostředí webu
Název v angličtině:	Object Manipulation Using Drag and Drop in the Web Environment
Rozsah práce:	36 stran + přílohy
Jazyk práce:	Český
Přílohy vázané v práci:	Seznam zkratk a pojmů, Obrázky, Tabulky, Zdrojový kód výukové aplikace, Zdrojové kódy pomocných aplikací
Klíčová slova:	táhni a pusť, JavaScript, webová aplikace, cloud, HTML5, CSS3, manipulace s objekty, výuka
Klíčová slova v angličtině:	Drag and Drop, JavaScript, Web Application, Cloud, HTML5, CSS3, Object Manipulation, Education

<p>Anotace práce:</p>	<p>Práce se skládá ze tří hlavních kapitol. První kapitola pojednává obecně o problematice techniky „táhni a pusť“ v prostředí webu, její historii a základních metodách k jejímu dosažení. Druhá kapitola je zaměřena na teoretické znalosti metody s využitím HTML5. Tyto znalosti by měly být dostatečné pro vytvoření aplikace, která techniku „táhni a pusť“ využívá. Poslední kapitola prezentuje praktickou část práce, ve které je uveden postup vytvoření výukové aplikace, která demonstruje možnosti praktického využití zvolených prostředků.</p>
<p>Anotace v angličtině:</p>	<p>The thesis consists of three main chapters. In the first chapter, there is generally described the problematic of Drag and Drop technique in the web environment, its history and the basic methods for implementing it. The second chapter is focused on the theoretical information about the HTML5 method. This information should be efficient to create an application which uses the Drag and Drop technique. The last chapter presents the practical part of the work, in which the process of creation of an educational application is described step by step and well commented. This application should demonstrate a usage of the chosen method.</p>