

Testování kapacity krátkodobé paměti (www.stackoverflow.cz)

Projekt ITU, 2019/20

Autoři: ~~xholas09~~, xplsek03, ~~xtrisk05~~

Zadání projektu

Cílové skupina

Relevantními uživateli webových aplikací jsou lidé ve věkovém rozmezí 15-60 let, a to kvůli zatížení uživatele při testování jinou činností. Test by neměly podstupovat osoby se srdečním onemocněním (plánovanou součástí testu bude vyvolání šoku). Test bude sloužit k porovnání kapacity krátkodobé paměti uživatele v kontextu jeho pracovní a věkové skupiny, čímž získáme informace k dalšímu výzkumu. Výsledky testu je třeba porovnávat relativně, protože neexistuje žádná metodika, která by určila absolutní výsledek.

Současné řešení a příčina problému

Testování a trénink krátkodobé paměti dnes uživatelé řeší zpravidla technologicky, na mobilních zařízeních je dostupná celá řada aplikací pro trénink krátkodobé paměti, existují i tréninkové internetové stránky [1] se zpravidla placenými testy. Pokud jde o příčinu problému, podle dr. Alloway spočívá v používání internetových a dalších médií, která obsahují velké množství rychle se střídajícího nesouvisejícího obsahu: YouTube, Twitter, psaní stručných SMS zpráv [2]. Podle jejího výzkumu tím dochází ke změnám pozornosti a dlouhodobě pak narušení růstu neurálních spojení, která jsou zodpovědná za přenos krátkodobých paměťových dat a jejich následné ukládání do dlouhodobé paměti [3]. Pokud jde o hry, podle Alloway rozvoji neurálních spojení prospívají: Tetris, pravidelná četba knih, pexeso, šachy (vůbec jakékoliv strategické a plánovací hry) a v neposlední řadě z oblasti médií dlouhodobé konverzace na sociálních sítích. Pokud jde čistě o trénink krátkodobé paměti, jedním z efektivních způsobů jsou mobilní i webové aplikace založené na N-back testování [4].

Naše řešení

Cílem projektu je vytvořit webovou aplikaci sloužící k testování krátkodobé paměti, která ale zároveň uživatele zatíží jinou myšlenkovou činností (přesná metodika bude popsána v následující části technické zprávy). Tento postup není u testování krátkodobé paměti obvyklý, dá se předpokládat, že obtížnost testu se tím zvýší. Existující aplikace se zpravidla zaměřují na trénink jedné oblasti (čísla, obrázky, tvary, pohyb), bylo by dobré tedy všechny tyto oblasti v měření zahrnout. Průběh testování uživatele bude následující: po zahájení testu bude vystaven cca. 10 úlohám z celého průřezu spektra běžných i neběžných

testovacích metod (zapamatování textu, obrázků, barev, pexeso, čísla, tvary, telefonní čísla, rozdíly ve velikosti objektů, pozice objektů, pořadí objektů, nákupní seznamy, loginy spolupracovníků na projektu ITU a IIS které jsem já sám udržel asi tři vteřiny). Před zadáním výsledku (nebo paralelně s veškerou svojí činností při testu) bude přerušován a nucen vykonávat jinou, se samotným testem nesouvisející činnost (odklikávání oken, vyvolání šoku, vyplnění svých osobních údajů, hudba na pozadí, simulovaný krátký telefonní hovor, labyrint pomocí myši, vyplnění reCAPTCHA formuláře, hry s klávesnicí [5] atd.). Na některé úlohy může být viditelně stanoveno časové omezení. Po ukončení testu dojde k relativnímu vyhodnocení výsledku v rámci svojí uživatelské skupiny a uložení výsledku pro potřebu dalších uživatel. Pokud se podaří navrhnout přehledné uživatelské rozhraní pro souběžné provádění testovací i nesouvisející zdržovací činnosti, můžeme očekávat zajímavé výsledky měření.

Zajímavé odkazy

[1]

<https://spomocnik.rvp.cz/clanek/11665/JAK-NA-KRATKODOBOU-PAMET.html>

[2]

<https://www.itpro.co.uk/614785/twitter-youtube-damage-working-memory>

[3]

https://cs.wikipedia.org/wiki/Baddeleyho_model_pracovni_pam%C3%AD_pam%C4%9Bti

[4]

<https://en.wikipedia.org/wiki/N-back>

[5]

<https://acelisweaven.github.io/arrow-hero/>

Návrh - GUI, backend

Technologie

Projekt je vytvářen v kombinaci webových technologií HTML/CSS/Javascript/PHP/MariaDB, bude použito standartního modelu MVC, na straně serveru bez použití zvláštních frameworků, na straně klienta pak s využitím javascriptového frameworku jQuery, pro CSS pak LESS/SASS a několika externích javascriptových knihoven s implementací, kterou použijeme při provádění testování uživatel (více podrobností v komentářích u síťového modelu).

Datový model

Datový model se nachází v příloze, manipuluje se zde s daty uživatele a v případě testů, u kterých je potřeba ukládat větší množství dat, i s daty, ze kterých se testy generují. Testová data mohou být jednorázově generována externími knihovnami (jedná se o obrázky, zvuky či text), nebo mohou být

tvořena obrázkovými daty uloženými na serveru, a v takovém případě se odkazy k obrázkům uloží do databáze na serveru (u některých testů mohou být obrázků stovky až tisíce). Jednotlivé testy jsou tvořeny průvodními texty a obrázky, uloženými v přímo v šablonách (view), případné negenerovatelné obsahy se pomocí controlleru na serveru vezmou z databáze (model). Operace specifické ke každému testu (vygenerované formuláře, dynamické skripty) jsou součástí view, jejich podrobnější funkce jsou popsány v komentářích u síťového modelu. Pokud jde o zadávání uživatelských dat, v průběhu testu bude subjekt postupně zadávat některé své údaje, důležité pro vygenerování podrobnějších přehledů (získáme údaje o skupinách, které často používají sociální sítě, TV, hrají hry nebo hrají na hudební nástroj atd.). Zadávání údajů v případě naší aplikace probíhá v rámci testování a je součástí (!) testu (jedna z prvních testovacích otázek, přerušení). Detaily k nalezení v síťovém modelu.

Návrh GUI

Základní testovací rozhraní (prvky, které je potřeba uživateli zobrazit při provádění testů) tvoří dvě šablony: u testu, kde se pracuje s pasivním přerušováním (situace, kdy je uživatel nucen soustředit se na jinou činnost a je odvrácena jeho pozornost od činnosti v testu) je to základní okno šířky 100%, které je v případě přerušení překryto dialoggovým oknem o stejné šířce: tím se obsah testu dočasně překryje. Aktivní přerušování (uživatel je nucen se zároveň s testem soustředit i na jinou činnost) je vyřešeno šablonou, kde obsah testu je v levém bloku vlevo (70%) a v pravém bloku (30%) je přerušující obsah. Součástí testu jsou 4 dialogová okna s bonusovými otázkami, jejichž odpovědi závisejí na testových otázkách. Tyto okna se zobrazí po otázkách 2, 4, 6, 8 a překryjí zbytek stránky. Odpovědní doba na tyto bonusové otázky je omezena na 7-10s. Všechny testy (testových otázek bude 9) jsou odlišného charakteru, stejně jako způsob přerušení u jednotlivých testů. Momentálně jsou testy optimálně navrženy, pokud jde o jejich pořadí a způsoby vyrušení u jednotlivých testů. Pořadí jednotlivých testů je následující:

| Test | Vyrušení | Gener. Bonusové otázky |
|---------------------|------------------------------|------------------------|
| 1: pexeso | Vyplnění uživatelských údajů | 1. uhodnutá dvojice |
| 2: slovo | Generování zvuků | |
| 3: barvy | Šok (zvuk + obrázek) | Posl. Zadaná barva |
| 4: Mizející objekty | 10 – 15 js oken | |
| 5: Tel. Číslo | Telefonní hovor | Poslední trojčíslí |
| 6: Nák. seznam | reCaptcha | |

| | | |
|--------------|---------------|------------------------|
| 7: ABCD test | bludiště | Odpověď na otázku č. 2 |
| 8: Obrázky | Hra s šipkami | |
| Tvary | Hra s prsty | |

Testové úlohy se přepínají automaticky po jejich splnění, test nelze ze zřejmých důvodů pozastavit.

K metodice hodnocení se zatím nebudu vyjadřovat. Počítám s tím, že ohodnocení jednotlivých úloh testu bude vynásobeno koeficientem, ale jakým, to ukáže až reálné testování na více uživatelích v další části projektu, které nám ukáže, jaká je skutečná obtížnost testu (pravděpodobně vysoká).

Komentář ke GUI a testování

Návrh GUI byl oběma testovanými subjekty přijat kladně. Na základě bouřlivě kladných emocí subjektu č. 2 (testování proběhlo v nočních hodinách v jistém podniku) byla vybrána .cz doména a hosting na wedos.com; subjekt se navíc domnívá, že by bylo vhodné po dokončení projektu poslat odkaz na aplikaci spolu s vhodným PR textem redaktorům iDNES - sekce Ona. Podle mého názoru se to dá vysvětlit pouze genialitou předloženého řešení. Diskuze ohledně projektu s částí týmu proběhla (druhá část nebyla lokálně přítomna), obě části týmu překvapilo, že ohledně GUI míříme velmi podobným směrem (návrh se v podstatě lišil v drobnostech jako časování a některé implementační aspekty testových otázek), což je prapděpodobně dobře. Komentáře k jednotlivým testům (stručný popis toho jak fungují) je k nalezení u každého testu. Z uživatelského průzkumu padnul návrh, aby se dynamicky obměňovalo pořadí jednotlivých testů a jednotlivé způsoby přerušení. Tuto možnost zvážím.

Bylo doporučeno rovněž ukládat data do profilu, a s ohledem na možnost dynamického generování nových a nových testů zobrazovat zlepšení v rámci uživatelského profilu. S možností implementace uživatelského profilu zatím nepočítám, pokud by k ní došlo, použily by se k registraci údaje ze sociálních sítí (Facebook social plugin): to by dále navazovalo na možnost sdílení testů v rámci svého profilu atd. Prozatím je klíčové vytvoření užitečného uživatelského rozhraní v rámci konkrétních testů.

Přílohy

1 - datový model

2 – nástin webové aplikace

3 – úloha Pexeso

4 – úloha Slovo

5 – úloha Barvy

6 – Mizející objekty

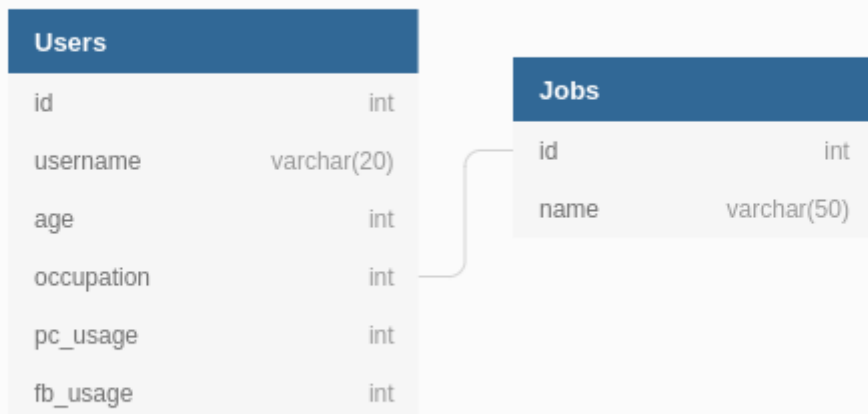
7 – Tel. Číslo

8 – Nákupní seznam

9 – ABCD test

10 – Obrázky

11 - Tvary



stackoverflow.cz
Otestuj si paměť.

Domů Test O projektu Kontakt

Hrubý nástin
šablony webové
aplikace.
Testová část bude
součástí bloku
Obsah stránky.

Obsah stránky

Lorem ipsum dolor sit amet, consectetur
adipisicing elit, sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua.

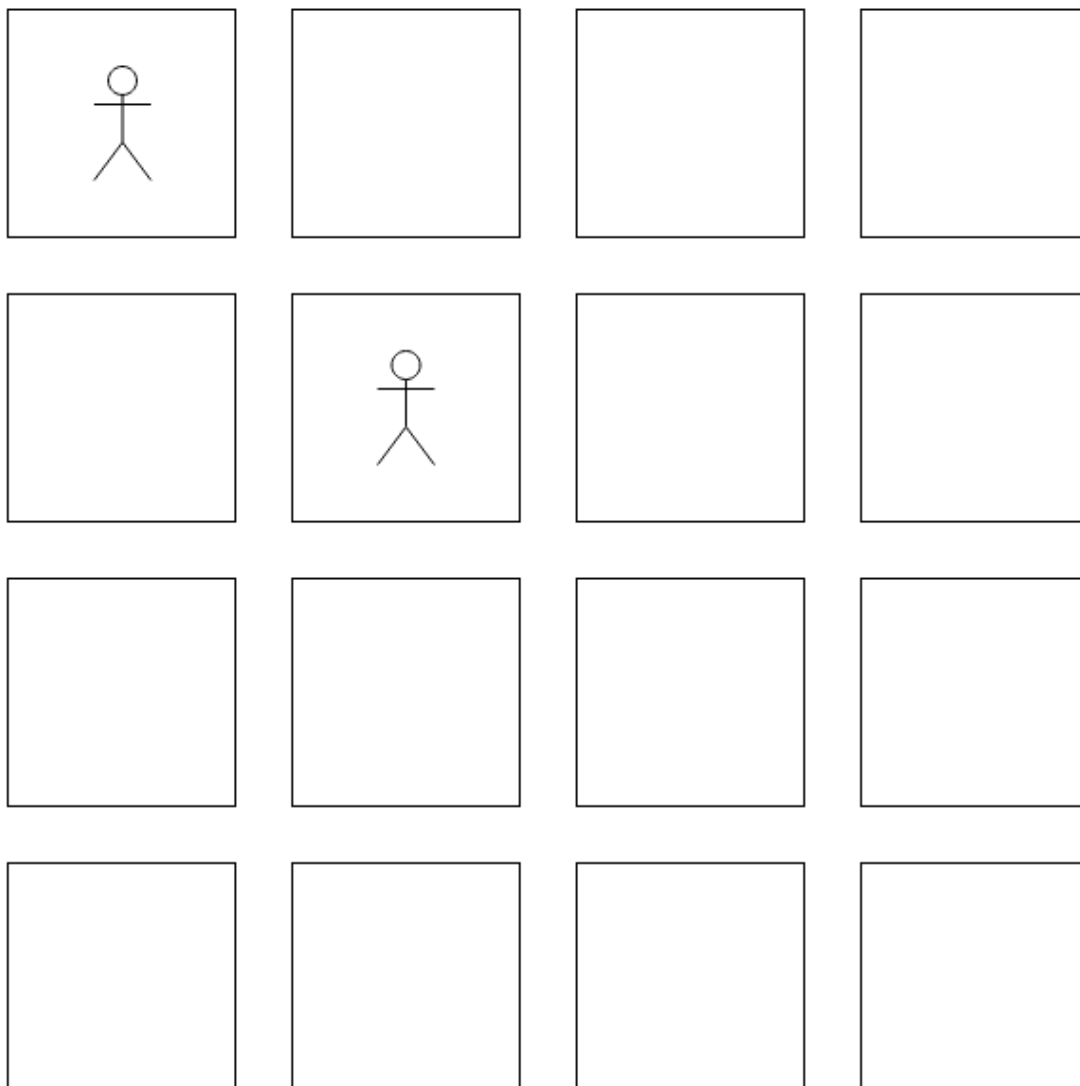
Go to test

Reklama 1

[Link to page >>](#)

Lorem ipsum dolor sit amet, consectetur
adipisicing elit, sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua.

Úloha 1/9 (Pexeso)



Pexeso 4x4

Na začátku hry nebudou uživatelům zobrazeny žádné pozice, půjde naslepo. Po každém tahu bude nucen do dialogu, který neprůhledně překryje celé okno, nucen zadat jeden z registračních údajů až do vyčerpání jejich počtu. Údaje jsou dostupné v datovém diagramu výše, jedná se např. o SELECT povolání nebo číselné údaje (1-5) ohledně používání sociálních sítí, PC, atd.

Úloha 2/9 (Slovo)

Zapamatujte si následující slovo:

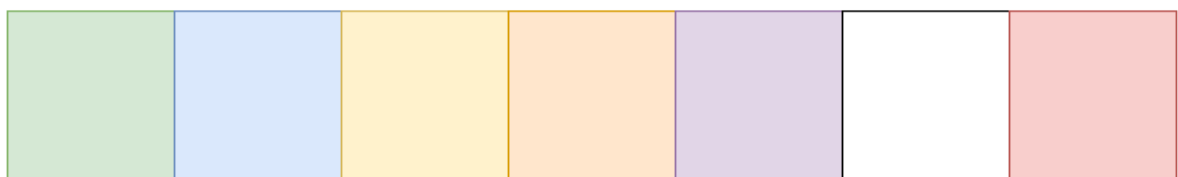
ZXGRTDNGGE

Zapamatování slova

Uživatel má 10s na zapamatování si slova složeného z náhodných písmen. Po 10s obrazovka ztmavne a po 1ř sekund se budou na pozadí generovat zvuky různých nepříjemných frekvencí. Po 15s zadá stejnou posloupnost písmen do formuláře a stiskem tlačítka přejde k další úloze.

Úloha 3/9 (Barvy)

Zapamatujte si následující barvy:



Zapamatování barvy

Uživatel má 10s na zapamatování sedmi náhodně vygenerovaných barev. Po 10s obrazovka ztmavne na 5s a následně se zobrazí obrázek a hlasitý zvuk. Poté uživatel musí setřídít bloky přetažením myši do původních pozic, bloky se promíchají. Setřídění potvrdí odesláním formuláře.

Úloha 4/9 (Mizející objekty)

Zapamatujte si pořadí, ve kterém objekty zmizely.

Zobrazit objekty

Mizející objekty

Uživatel kliknutím na tlačítko zobrazí animaci, ve které mizí 3x3 objektů v určitém pořadí. Po zmizení objektů se zobrazí mřížka 3x3 objektů a uživatel na ně musí kliknout v pořadí, ve kterém zmizely. Mezi ukončením animace a zobrazením mřížky vyskočí 10-15 JS oken v prohlížeči s textem "čtěte pozorně!" atd.

Úloha 5/9 (Telefonní číslo)

Zapamatujte si telefonní číslo:

+420 731 739 666

+ 43 6653 444 54

Calling..

Přejetím po displeji
přijmete hovor

Telefonní číslo

Uživatel má 7-10s na zapamatování telefonního čísla, poté se v pravé části stránky zobrazí blok s příchozím hovorem. Uživatel hovor "zvedne" přejetím jQuery slideru směrem doleva, poté se levý blok s číslem stáhne na width=0% a pravý na width=100%. Uživateli se přehraje předpřipravený soubor s hovorem, ve kterém je instruován k zapamatování 2-3 kratších vět. Zpráva se několikrát zopakuje a hovor se ukončí. Poté je uživatel vyzván k zadání původního telefonního čísla v levém bloku s width=100%.

Úloha 6/9 (Nákupní seznam)

Zapamatujte si nákupní seznam:

Žlutá ponožka 1.50 €
Vynikající kornflejky 4 €
Žlutá ponožka 1.50 €
Vynikající kornflejky 4 €

Nákupní seznam

Uživatel má 15-20s na zapamatování si seznamu o cca 8 položkách. Poté se zobrazí překrývací dialog s vyplněním reCaptcha formuláře, který 2x selže a vyplní se až napotřetí. Poté se zobrazí ca 50 položek v seznamu a uživatel musí kliknout na 8 položek ze seznamu které byly v původním seznamu.

pozn. K nákupnímu seznamu: názvy položek a ceny se generují náhodně.

Úloha 7/9 (Test)

Jaká je barva vody?

☐ ☒ ☐ ☐

Jaká je barva vodky?

☐ ☐ ☒ ☐

Jaká je barva?

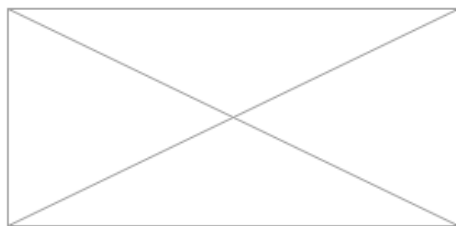
☐ ☒ ☐ ☐

Test

Uživatel má 20s na zapamatování si testových ABCD odpovědí u cca 8 - 10 otázek. Po uplynutí času se zobrazí dialog se středně obtížným bludištěm, které uživatel musí projít myší. Po dokončení bludiště dialog zmizí a uživatel musí vyplnit k testovým otázkám správné odpovědi. Jedna otázka může mít pouze jednu správnou odpověď. Potvrdí odesláním formuláře. Otázky i odpovědi se generují náhodně.

Úloha 8/9 (Obrázky)

Pokud už se obrázek jednou objevil, stiskněte SPACE.



Obrázek

Úloha se spustí kliknutím na tlačítko start, po kliknutí se v intervalu 3s zobrazí obrázek z databáze. Bude se jednat o obecné obrázky krajin apod. Dohromady se zobrazí cca 20-25 obrázků, pokud už se obrázek někdy objevil, uživatel musí v intervalu 1-2s po dobu zobrazení obrázku stisknout SPACE. Zároveň se v pravém bloku stránky zespodu objevují šipky v intervalu 1.5s zobrazují šipky, který vyjždí směrem nahoru v bloku. Uživatel musí v příslušném okamžiku stisknout na klávesnici odpovídající šipku. Úloha skončí automaticky.

Úloha 9/9 (Tvary)

Tvary

Na obrazovce se zobrazí cca 4x4 vygenerovaných obrázků, mezi kterými jsou minimální rozdíly. Budou na obrazovce cca 10s, pak se po dobu 1s označí jeden obrázek (změní se jeho pozadí na zelenou), obrazovka se zatmaví na 15-20s. Uživatel je v mezechase vybidnut, aby si zkusil hru s prsty (zajímavý experiment, kdy je potřeba na jedné ruce zvednout ukazovák a na druhé palec, a následně je v rychlém intervalu prostrídat). Po nějaké době se zobrazí stejná mřížka, akorát proházená a uživatel musí kliknutím určit zvýrazněný tvar.

Ihned po ukončení poslední úlohy je vypočítán rating a zobrazeno celkové porovnání v rámci uživatelské skupiny (povolání, věk, atd.).

Architektura aplikace a Návrh testování

Zvolené technologie a architektura

Pro test jsem se rozhodl použít MVC, pro model databázi MySQL, pro controller PHP a pro view HTML/CSS/JS/Jquery. Pokud jde o uživatelské rozhraní, používám na straně klienta jak čistý JavaScript pro generování různých časových událostí, tak knihovnu Jquery pro práci s objekty v DOM, a v neposlední řadě pro některé prvky GUI knihovnu JQueryUI a pro validaci formulářů na straně klienta knihovnu JQuery Validate (<https://jqueryvalidation.org/>). JQuery používám hlavně kvůli snadnému přístupu k objektům DOMu a usnadněné práci s nimi (funkce pro hledání prvků, nalezení jejich pořadí, indexu apod.), potom také kvůli funkci `animate()` využívané v některých "pohyblivých" testech. JQuery Validate používám kvůli jednoduché vestavěné validaci typů polí a proto, že dokáže dynamicky z kódu odeslat formulář zvoláním funkce `submit()`. Technologie jsou zvoleny proto, že ve většině případů tvoří testy sled přesně načasovaných událostí (JS funkce `setTimeout`, `setInterval`), které provádějí předem dané operace v daném čase, zpravidla se jedná o zobrazování/skrývání/přidávání DOM elementů.

MVC a napojení

Na straně serveru jsou pro každý test jednotlivě uložena testová data, ze kterých se dané testy krátkodobě paměti generují. Některá data jako odkazy na obrázky nebo zvukové soubory testu jsou uložena v MySQL databázi, pro každý test jsou uloženy šablony, které se vkládají do hlavní testové kostry, společné pro všechny jednotlivé testy. Hlavní kostru tvoří jednoduchá šablona s hlavičkou, obsahem testovacího okna a patičkou. Hlavní testovací obsah (telefonní číslo, náhodná slova, fráze apod.) se generují v controlleru a vkládají se dynamicky do šablony, odkud si je pak javascript sám načte z konkrétního DOM elementu. Pokud jde o dynamické chování testů, všechny dynamické operace, které probíhají na straně klienta, jsou uloženy v samostatném JS souboru, na který se odkazuje v hlavičce konkrétní stránky testu. Soubory jsou rozděleny, protože dynamických funkcí je pro všechny testy relativně hodně a nemá smysl načítat je všechny zároveň. Průměrně má JS soubor 200+ řádků kódu, vzhledem k jednoduchosti testů a předem známé posloupnosti transakcí a operací je použito globálních proměnných, se kterými funkce testu pracují. Zpravidla kód souboru tvoří globální proměnné, funkce a poté statement `$(document).ready`, který obsahuje posloupnost operací, které se provedou po načtení DOM. Kvůli zjevné jednoduchosti provádění testů (viz. Průběh testů) nebylo potřeba nějaké složité třídy, mezivýsledky testů se dočasně ukládají do databáze a na konci testu jsou trvale zapsány jako konečný výsledek.

Průběh testů

Jednotlivé testy probíhají tak, že po přečtení úvodního dialogu ke každému testu,

kde je napsané, o co v testu půjde a jaké jsou požadované klientské operace, se spustí posloupnost načasovaných funkcí, které na určitou dobu zobrazí obsah k zapamatování a následně spustí některé z přerušení (může být prováděné současně s testem, nebo může být test tímto přerušením pozastaven).

Uživatelská interakce s přerušením testu (buď nezávislá - samostatně časovaná, nebo spoléhající na některou uživatelskou operaci - klikání myší, psaní na klávesnici..) vyvolá posloupnost dalších operací, zpravidla zobrazení testového okna s formulářem zadání zapamatovaného výsledku. Výsledek testu, který se po validaci formuláře (Jquery Validate) odešle a který PHP skript zpracuje, je průběžně uložen a je použit při závěrečném vyhodnocení testu. Po uložení mezivýsledku testu je controllerem vygenerován další test v pořadí a odeslán ke klientovi. Testy se hodnotí pouze jako celek, tj. skóre všech osmi testů, ale ne odděleně!

Pozn.: jedne z testů byl vypuštěn kvůli celkové délce celého testování, která by neměla přesáhnout 18 minut.

Datový model

Protože není potřeba ukládat u každého uživatele uživatelské údaje (pro účely srovnání jednotlivých uživatel má smysl ukládat pouze údaje, které uživatele charakterizují - používání sociálních sítí, věk atd.) - stačí výsledek testu se všemi charakteristikami uložit do jedné tabulky, včetně výsledku testu a pouze zaznamenat, jestli je výsledek testu finální. Nedokončené testy starší než 24 hodin se budou z databáze průběžně odstraňovat automaticky. ER diagram z druhé části zůstane v podstatě nezměněn, jen s triviálními odlišnostmi (bool final_result označující konečné skóre testu apod.).

Testovací otázky

Z hlediska zaměření projektu je stanovení testovacích otázek zřejmé: "Testuje projekt jako celek ukládání informací do krátkodobé paměti?" a "Je ukládání paměti do krátkodobé paměti efektivně narušováno v průběhu testu?", a v neposlední řadě "Není ukládání informací do krátkodobé paměti přerušováno až příliš?" a také "Je uživatelské rozhraní v průběhu testu natolik intuitivní, aby získané výsledky byly narušované pouze cíleným přerušováním v průběhu testování?". Vzhledem k obecnému zaměření z hlediska cílové skupiny bude nejefektivnější provést pro stanovení optimální náročnosti testu a optimálnosti GUI (tj. odpověď na testovací otázky 1-4) testování s přímým pozorováním, a provést ho na několika subjektech v průřezu všech věkových kategorií a obecně oborového zaměření (prostě řečeno = kdokoliv). Po analýze výsledků testování (klidně několikanásobném provedení testování) může být projekt "vypuštěn" pro další testování, tentokrát už bez přímého pozorování, pouze se zpětnou vazbou - pravděpodobně dialogové okno s formulářem, které se otevře po kliknutí na odkaz umístěný v záhlaví stránky, a přes který bude možné odeslat feedback k testu. Pravděpodobně se tak stane zveřejněním odkazu na sociálních sítích, s krátkou prosbou o účastnění se testování, které je pojato jako značně "netradiční". Po provedení sekundárního testování budou feedbacky náležitě

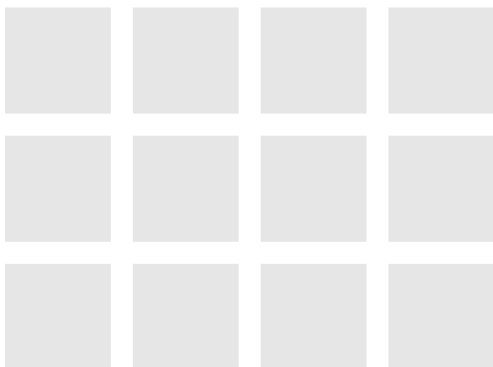
zpracovány. Pozn.: jelikož se jedná o webovou aplikaci, předpokládá se, že nedílnou částí feedbacků budou tvořit bugy v rozdílném zobrazování aplikace na různých verzích prohlížečů a zařízení. Testování samo o sobě musí být každým uživatelem provedeno až dokonce, tj. musí absolvovat všech osm testů. V případě sekundárního testování nám to umožní začít stavět vyhodnocovací bázi výsledků (v případě primárního testování by byly výsledky narušovány možnými hrubými technickými chybami a bugy, které už se při sekundárním testování neočekávají). Testování krátkodobé paměti nemá žádnou relativní vyhodnocovací stupnici a pro interpretaci výsledku je třeba mít data = relativní porovnání. Projekt není určen - kvůli povaze několika testů - pro mobilní zařízení!

Proběhlé testování a výsledky

Prozatím probíhalo izolované testování na několika subjektech, zpětná vazba již byla zpracována do testů jak jsou zobrazeny v tomto dokumentu. Jendalo se převážně o časování akcí jako zobrazování, mizení a délka trvání některých přerušení; "think-loud protocol" bylo tedy prováděno do jisté míry. Protože testy se musí provádět podle předem daného scénáře, samotné testování nedává uživateli prostor pro vlastní akce. Rozlišujeme ovšem dva druhy výsledku testování: v primární fázi chceme, aby uživatel prováděl operace, po kterých se test bude chovat nestandardně (chceme aby "hacknul" JS kód: klikal opakovaně na tlačítka provádějící uživatelské operace, naboural časování operací, zadával chybné vstupy nebo je nezadával vůbec atd.), v druhé části už jde primárně o sběr dat výsledků pro uložení do databáze a nahlášení bugů v různých prohlížečích. Prozatím byly implementačně zpracovány některé bugy v jednotlivých testech, chybné vstupy formulářů apod.

Náhledy některých vybraných testů

Pro představu o tom, jak vypadá test a jeho GUI. Dialogy, některá přerušení testů a obsah vybraných testů.



1 / 9

Test 1: pexeso



Uzivatel'ske Jmeno

Heslo

Potvrzeni hesla

Pokracuj

1 / 9

Test 1: přerušení uživatele zadáváním údajů, dialog



LOrEMipsUM

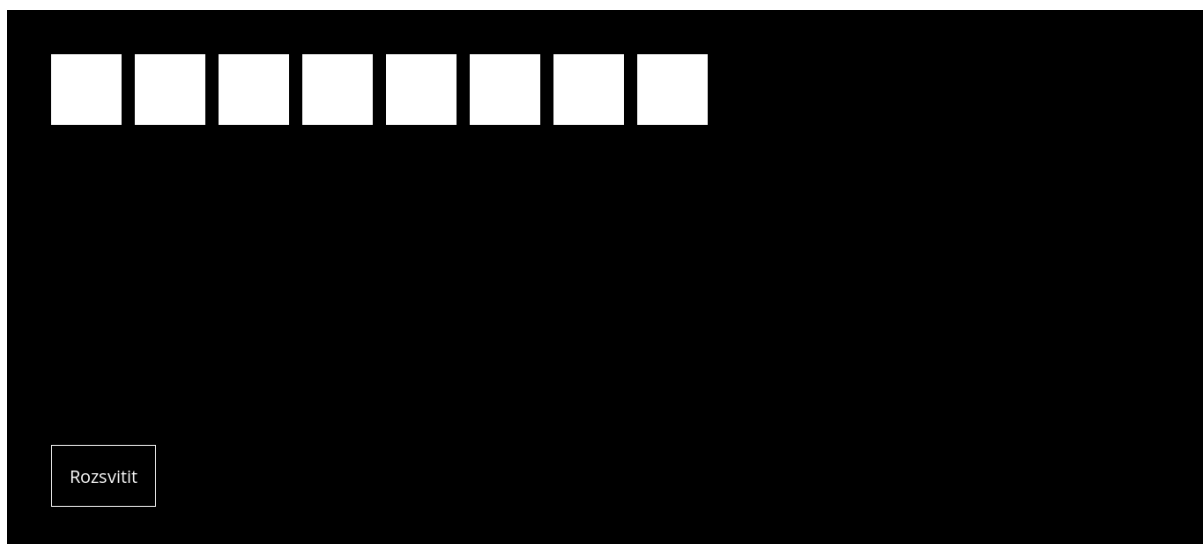
2 / 9

Test 2: zapamatování slova

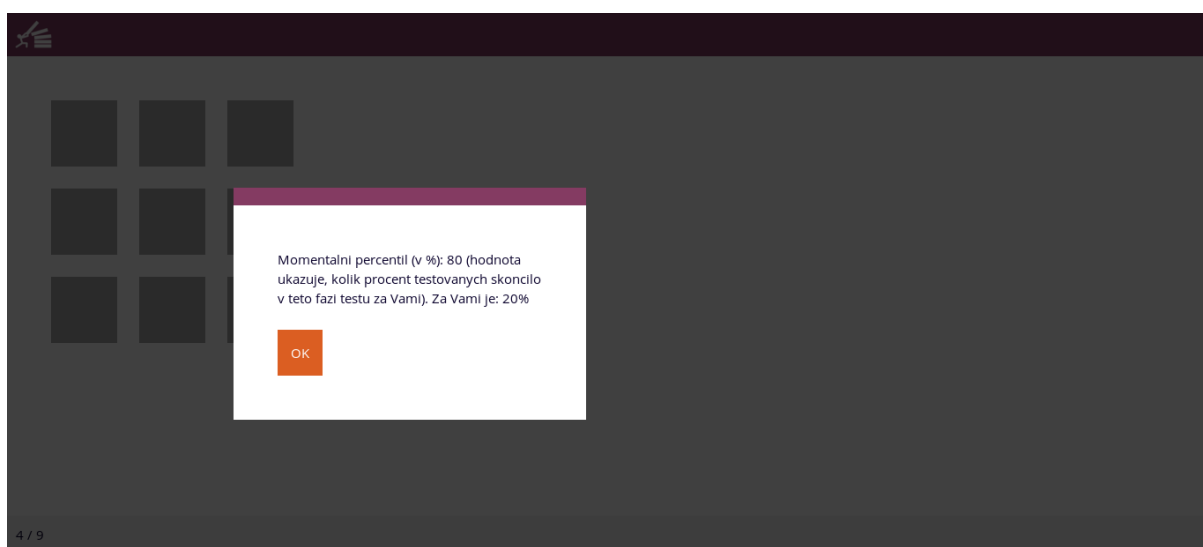


3 / 9

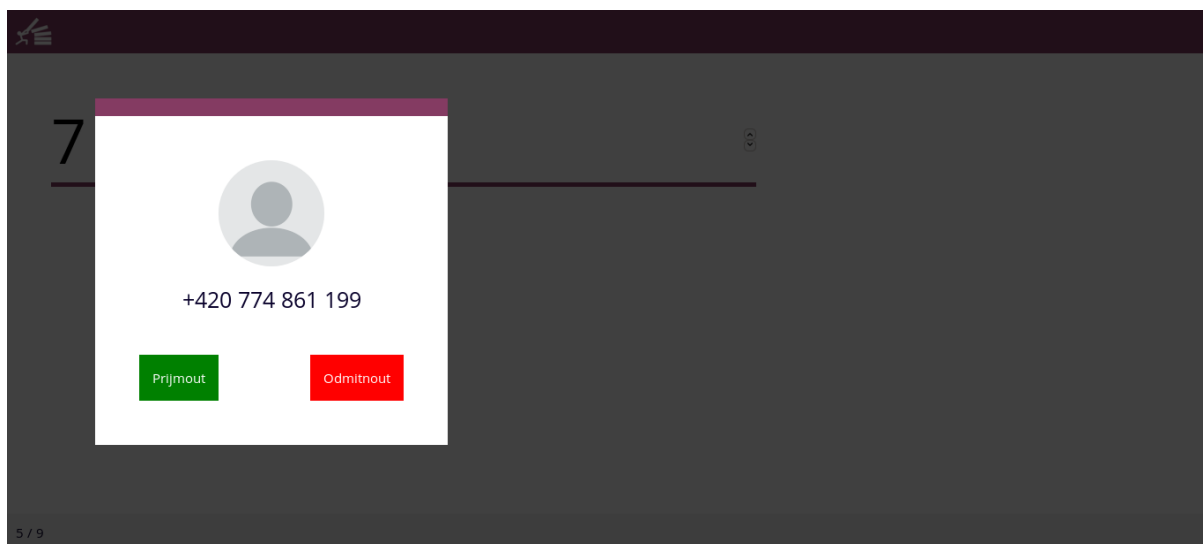
Test 3: zapamatování barev



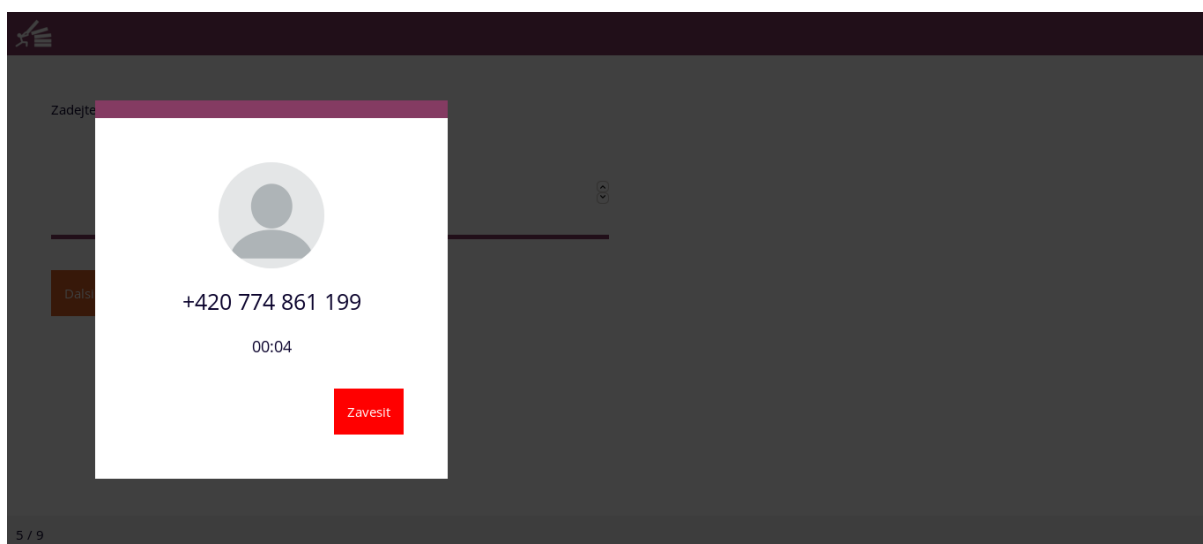
Jedno z přerušení u barev



Přerušení u zapamatování pořadí mizení bloků



Přerušení u testu zapamatování tel. čísla



Přerušení u testu zapamatování tel. čísla



LEDL, maloobchodní prodejna
Falesna 1, Brno-Lesna
id účtenky: 456743678976
datum a čas prodeje: 15/4/2019 5:25:25
id pro EET: g0t0h411y0u455

| | |
|---------------------------|------|
| dovolena v karibiku | 89€ |
| parkova oslava | 35€ |
| zelena zirafa | 150€ |
| linz | 689€ |
| odkladac | 555€ |
| tri muzi ve clunu | 6€ |
| boxovací pytel | 50€ |

celkova cena: 1574€
celkem zaplaceno: 0€
Dekujeme Vam za nakup!

6 / 9

účtenka k zapamatování



7



8 / 8

Test 8: slideshow opakovaných obrázků přerušované hrou „Arrow hero“

Implementace a výsledky testování

Podstatné části implementace

Serverový kód byl vyřešen jako sada cca. 10 php skriptů, které jsou propojeny formuláři, odkazující na další skript v pořadí. Šablony jsou implementovány po částech, aby se kód neopakoval, a jsou vkládány pomocí funkce *include*. Úvodní stránka obsahuje proklik na první z testů, který sesbírá základní uživatelské údaje (je součástí testu, viz kapitola o obsahu testů) a kliknutím na odeslání výsledku testu odešle ve výsledek ke kontrole testu číslo dvě. Mezivýsledky testu jsou ukládány do proměnné `$_SESSION`, která je na serverové straně nastavená na výchozích 24 minut. V případě že některý zadaný údaj není validní (což naznačuje pokus o hacking, protože vstupy jsou testovány pomocí Jquery validate pluginu i na klientské straně), dočasné údaje se vymažou a uživatel je odkázán na začátek testu. V případě úspěšného průchodu testem je zobrazen výsledek testu se zaměřením na relativní porovnání s ostatními uživateli (se všemi a z jeho zadané skupiny zvlášť). Session proměnná se poté vymaže a zničí.

MySQL databáze běží u poskytovatele webhostingu a správa je prováděna přes phpMyAdmin. Back-end není implementován, nebyl pro něj důvod (případné nevalidní údaje z databáze je možné promazávat ručně jednou za měsíc, což by byla jediná funkce back-endu).

Uživatelské testování – fáze 1

První fáze testování byla provedena v rámci implementace na několika subjektech za účelem doladění časů některých přerušení testů a taky zjištění, jak přílišný odpor test u subjektů vyvolá. Výsledky v obou případech rozhodně nezklamaly, navíc byla přidána i validace k některým formulářům testů a přidána validace formulářů na straně serveru. V první fázi na každém testu proběhly dva průchody za použití think-loud, závěry byly zapracovány do implementace (přesné výsledky už si nepamatuju, ale byly zabudovány do implementace tak jak je popsána). Každý test byl „odblokován“, dalo se k jeho skriptu dostat odděleně, a bylo nejprve ověřováno zadávání nevalidních vstupů uživatelem a posléze validních vstupů. Hodnoty byly následně pro ověření vypsány funkcí `var_dump()` do stránky. Hlavní „neviditelná chyba“, která byla tímto odhalena, byl špatný způsob validace vstupního pole typu checkbox, kde byl špatně použitý vykřičník v serverovém skriptu. Každopádně byla odhalena řada drobných bugů, nejvíce z hlediska nekompatibility mezi různými prohlížeči. Kompatibilita byla stanovena na novější verze prohlížečů Firefox/Chrome, pro prohlížeče s jádrem IE není kompatibilita zaručena (klíčové prvky testu by nicméně fungovat měly).

Průchod testem dává smysl pouze jednou! Uživatel podruhé ví, jaké a kdy má očekávat úlohy a vyrušení.

Uživatelské testování – fáze 2

Poté co byl systém odladěn prvotním testováním, mohlo se přejít k dalšímu kroku, a to samostatnému uživatelskému testování. Spolupracující (někdy i dobrovolně) uživatelé byli vyzváni k pevně definované úloze, což je postupný průchod testem (v tomto případě neměl volný průchod smysl, všechny serverové skripty jsou provázány tak, aby se k testu uživatel dostal pouze z předchozího testu, tedy až na 1. test). Zadávané údaje měly být validní a tato fáze měla simulovat ostrý provoz. Testování už neodhalilo nijak závažné chyby, kromě chybějící validace jednoho formulářového pole na straně klienta. Testování mělo zároveň sloužit ke sběru porovnávacích výsledkových dat, bohužel byla vlivem mé stupidity databáze narušena a posléze musela být uvedena do výchozího stavu.

Původně bylo v plánu i rozšířené testování se zapojením většího množství subjektů, ale nezbyl čas. Testovací stránka byla doplněna o tlačítko „Nahlásit bug“, který v nové záložce otevřel google formulář na nahlášení chyby.

Výsledky testování a odpovědi na otázky

V části o testování dokument definoval testovací otázky:

"Testuje projekt jako celek ukládání informací do krátkodobé paměti?"

"Je ukládání paměti do krátkodobé paměti efektivně narušováno v průběhu testu?"

"Není ukládání informací do krátkodobé paměti přerušováno až příliš?"

"Je uživatelské rozhraní v průběhu testu natolik intuitivní, aby získané výsledky byly narušované pouze cíleným přerušováním v průběhu testování?"

Odpověď na otázku 1 vyplynula z první fáze testování, některé úlohy více a některé méně. Cílem úloh je narušit pozornost uživatele, ale to nakolik se uživatel soustředí nebo co vyruší je dost subjektivní.

Uživatelské rozhraní se ukázalo být intuitivní, některé přechody a vyrušení sice uživatele překvapily, ale test plyne dopředu sám od sebe: nebyla potřeba žádná asistence při průchodu testem.

Odpověď na otázku 2 a 3: opět některé testy narušují pozornost více a některé méně, subjektivní záležitost. Testy jsou schválně navrženy tak aby byly těžké (některé i porušují pravidlo 10s na uložení dat v krátkodobé paměti), ale ukázalo se že při použití různých mnemotechnických pomůcek se testová data dají zapamatovat.

Ukázalo se, že část uživatel vůbec nečte textové popisy úloh a pak jsou něčím překvapení. Z dalšího testování bychom poznali (což by mě dost zajímalo), jestli je zvýraznění textu donutí tyto popisy číst.

Závěr

Jak se (opět) ukázalo, testování je nedílnou součástí vývoje každé aplikace. Pokud je provedeno správně, vede nejen k opravě chyb, ale i k vyšší efektivitě při plnění cílů, které stanovuje dokumentace. Uživatel občas chce něco, co by vývojáře nikdy ani nenapadlo.

Nasazeno do provozu

Aplikace běží na www.stackoverflow.cz.

Práce v týmu a nové zkušenosti

Pokud jde o mě, pracuji zpravidla samostatně, takže práce v týmu byla pro mě nezvyklá a doufám, že ji nebudu muset v dohledné době opakovat. Dopadlo to přesně jak jsem očekával, jeden člen nedostal zápočet a druhý člen noc před odevzdáním prohlásil, že na to nemá čas. Veškerý odevzdaný kód je tedy autorský. Kolegu co nedostal zápočet chápu, dal vědět dlouho dopředu. Toho druhého nehodlám komentovat.

Naučil jsem se zajímavé věci s jQuery, takže práce byla v tomto směru přínosem.

Přílohy

Jsou součástí jednotlivých kapitol pro lepší přehlednost, z textu se na ně zpravidla přímo odkazuje.

