

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Software pro prezentaci stimulů

Plzeň, 2013

Miroslav Vozábal

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

.....

V Plzni dne 5. 5. 2013, Miroslav Vozábal

Poděkování

Tímto bych chtěl poděkovat vedoucímu práce Ing. Romanovi Moučkovi, Ph.D. a Ing. Petru Jarošovi za jejich rady a čas, který mi věnovali při řešení dané problematiky.

Abstract

The Neuroscience Stimuli Presentation Software

This bachelor project is dedicated to improve the functionality of the software PreSti. The software is designed for the presentation of stimuli in the experiments based on the method of evoked potentials (ERP). Currently, various commercial software exist in this area. The medicine field attracts the largest attention of PreSti since there is a shortage of experts who can operate this software. Most of the current software use graphical interface or even their own programming language when creating and presenting stimuli. PreSti allows for an additional way of creating stimuli through visual programming. The visual programming language of PreSti enables healthcare professionals to create stimuli in simpler ways and offers a sufficient flexibility in creating test scenarios. Although Presti visual programming seems to be more applicable in creating test scenarios, it is not usable without the knowledge of the visual programming language. The aim of this bachelor project is to solve this drawback by creating a user support (a guide) allowing for the design of test scenarios.

Abstrakt

Software pro prezentaci stimulů

Tato bakalářská práce se věnuje vylepšení funkčnosti softwaru PreSti. Software je určen pro prezentaci stimulů v experimentech využívajících metodu evokovaných potenciálů (ERP). V současnosti existují v této oblasti různé komerční softwary. Největší využití mají tyto softwary v oblasti medicíny, kde se zřídka nacházejí odborníci, kteří umí programovat. Většina těchto softwarů využívá pro vytváření a prezentaci stimulů grafické rozhraní, nebo dokonce vlastní programovací jazyk. PreSti přichází s jiným způsobem vytváření stimulů, a to prostřednictvím vizuálního programování. Vizuální programovací jazyk PreSti umožňuje pracovníkům ve zdravotnictví vytvářet stimuly jednodušším způsobem, a to i s dostatečnou flexibilitou při vytváření testovacích scénářů. Ačkoli vizuální programování v PreSti se zdá být vhodnější způsob vytváření testovacích scénářů, neřeší možnost vytvořit scénář bez znalosti vizuálního programovacího jazyka. Cílem této bakalářské práce je odstranit tento problém vytvořením uživatelské podpory (průvodce) pro vytváření testovacích scénářů.

Obsah

1	Úvod	1
2	Evokované potenciály	2
2.1	ERP experimenty	2
2.2	Měření EEG signálu	2
2.3	Měření ERP experimentů	3
2.4	Vyhodnocení ERP experimentu	4
3	Software pro prezentaci stimulů.....	5
3.1	Seznam softwarů pro prezentaci stimulů.....	5
3.1.1	DirectRT	5
3.1.2	Presentation	7
3.1.3	Paradigm.....	9
3.1.4	PreSti	11
3.2	Porovnání softwarů.....	13
3.2.1	Funkcionalita	13
4	Specifikace požadavků	15
4.1	Úvod	15
4.1.1	Rozsah práce	15
4.2	Obecný popis	15
4.2.1	Kontext produktu.....	15
4.2.2	Funkce produktu.....	15
4.2.3	Třídy uživatelů	15
4.2.4	Omezení	16
4.3	Specifikační požadavky	16
4.3.1	Funkce	16
4.3.2	Požadavky na údržbu	16
4.3.3	Licenční podmínky.....	16
4.3.4	Autorská práva	16
5	Návrh a implementace.....	17
5.1	Vizuální programovací jazyk.....	17
5.1.1	Definice vizuálního programovacího jazyka.....	17
5.2	Základní testovací scénář.....	17

5.2.1	Upravitelnost testovacího scénáře vygenerovaného průvodcem	17
5.2.2	Jednoduchost vytvoření testovacího scénáře pomocí průvodce.....	17
5.3	Kontext systému	18
5.4	Architektura systému	19
5.4.1	Původní sada komponent	19
5.4.2	Rozšíření sady o komponentu PreStiGuide.....	20
5.5	Implementace.....	21
5.5.1	Průvodce.....	21
5.5.2	Návrh tříd	23
5.5.3	Dodatečné mimofunkční úpravy PreSti	26
6	Testování	27
6.1	Pravděpodobnostní průvodce	27
6.1.1	Vytvoření testovacího scénáře.....	27
6.1.2	Zhodnocení funkčnosti testovacího scénáře.....	28
6.2	Poziční průvodce	29
6.2.1	Zhodnocení funkčnosti testovacího scénáře.....	30
6.3	Použité symboly	30
6.4	Otestování uživatelského rozhraní uživateli.....	30
6.4.1	Průběh testování	31
6.4.2	Výsledky testování	31
7	Zhodnocení.....	32
7.1	Návrhy na budoucí vylepšení	32
7.1.1	Interstimulus interval.....	32
7.1.2	Symbol uchovávající hodnoty	32
7.1.3	Úprava symbolu RandomGenerator.....	33
7.1.4	SoundStimulus	33
7.1.5	Log soubor.....	33
7.1.6	Chyba při navazování spojení symbolem Trigger.....	33
8	Závěr.....	34

1 Úvod

Tato bakalářská práce se věnuje vylepšení funkčnosti softwaru PreSti, který byl původně vytvořen Petrem Jarošem [6]. Software je určen pro prezentaci stimulů v experimentech využívajících metodu evokovaných potenciálů (ERP) (viz 3). V současnosti existují v této oblasti různé komerční softwary. Největší využití mají tyto softwary v oblasti medicíny, kde se zřídka nacházejí odborníci, kteří umí programovat.

Většina softwarů využívá pro vytváření a prezentaci stimulů grafické rozhraní nebo dokonce vlastní programovací jazyk. PreSti přichází s jiným způsobem vytváření stimulů, a to prostřednictvím vizuálního programování. Vizuální programovací jazyk (viz 5.1) PreSti umožňuje pracovníkům ve zdravotnictví vytvářet stimuly jednodušším způsobem, a to i s dostatečnou flexibilitou při vytváření testovacích scénářů.

Ačkoli vizuální programování v PreSti se zdá být vhodnější způsob vytváření testovacích scénářů, neřeší možnost vytvořit scénář bez znalosti vizuálního programovacího jazyka. Cílem této práce je odstranit tento problém vytvořením uživatelské podpory (průvodce) pro vytváření testovacích scénářů.

V teoretické části je čtenář nejprve seznámen s pojmy, které se vztahují k problematice evokovaných potenciálů (viz 2). Druhá část se zabývá popisem softwarů pro prezentaci stimulů a jejich následným porovnáním.

V úvodu praktické části jsou specifikovány požadavky na novou funkcionalitu. Na základě požadavků je vytvořen návrh. Následuje popis implementace daného návrhu a popis dalších úprav softwaru PreSti. V závěrečné části je nově přidaná funkcionalita otestována a je vytvořen seznam dalších možných vylepšení softwaru.

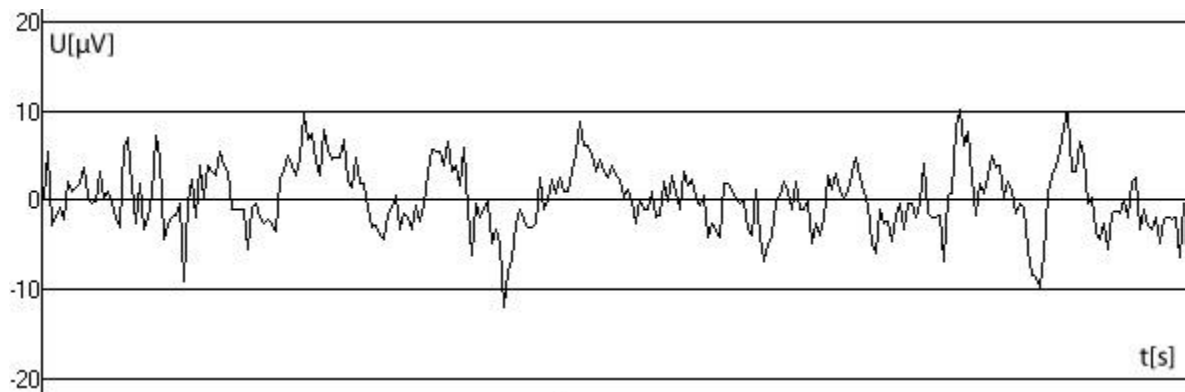
2 Evokované potenciály

2.1 ERP experimenty

ERP (Evokovaný potenciál, Event Related Potential) je reakce mozku na vnější a vnitřní podněty. Tyto podněty vyvolávají změnu elektrického napětí uvnitř mozku. Elektrická aktivita (EEG signál) je měřena neinvazivně na skalpu subjektu. Při stimulování mozku různými podněty se snažíme potlačit podněty, které s experimentem nesouvisí, protože by mohly ovlivnit jeho výsledek. Účelem těchto experimentů je vyhodnocení reakce mozku na stimuly, pochopení jednotlivých složek EEG signálu a také lokalizace části mozku, která je zodpovědná za zpracování stimulu. [6]

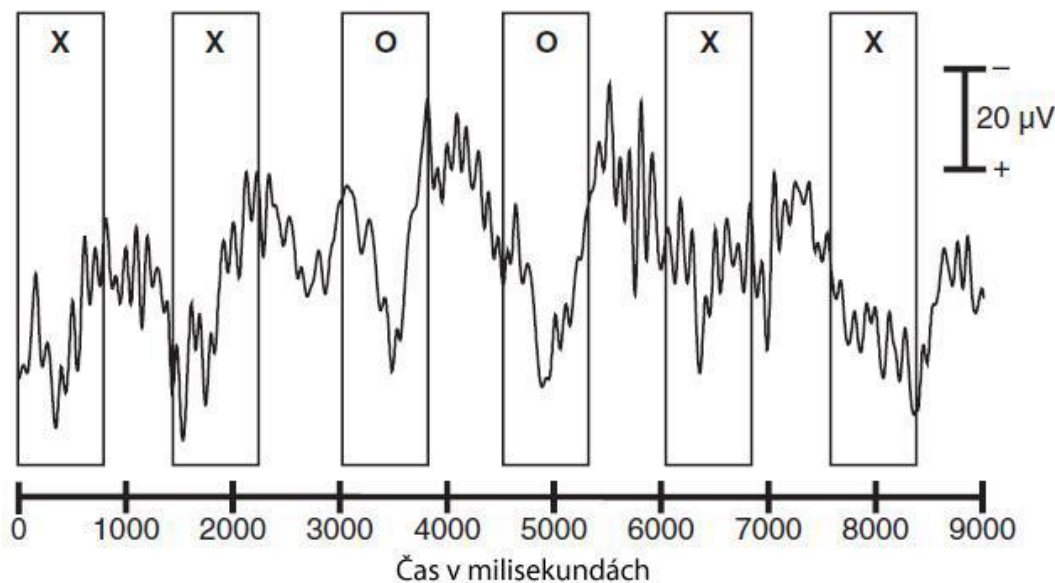
2.2 Měření EEG signálu

„EEG signál (neboli elektroencefalogram) je záznam časové změny elektrického potenciálu způsobeného mozkovou aktivitou [1]. Měří se neinvazivně na skalpu objektu elektrodami, které jsou na něm rozmístěny. Jedna z elektrod je hlavní (referenční). Měřené signály jsou rozdíly potenciálů jednotlivých elektrod vůči elektrodě referenční. Na obrázku 2.1 je vidět typický průběh EEG signálu.“ [6]



Obr 2. 1 Příklad EEG signálu (Signál odpovídá aktivitě odpočívajícího subjektu s otevřenými očima) [2].

2.3 Měření ERP experimentů

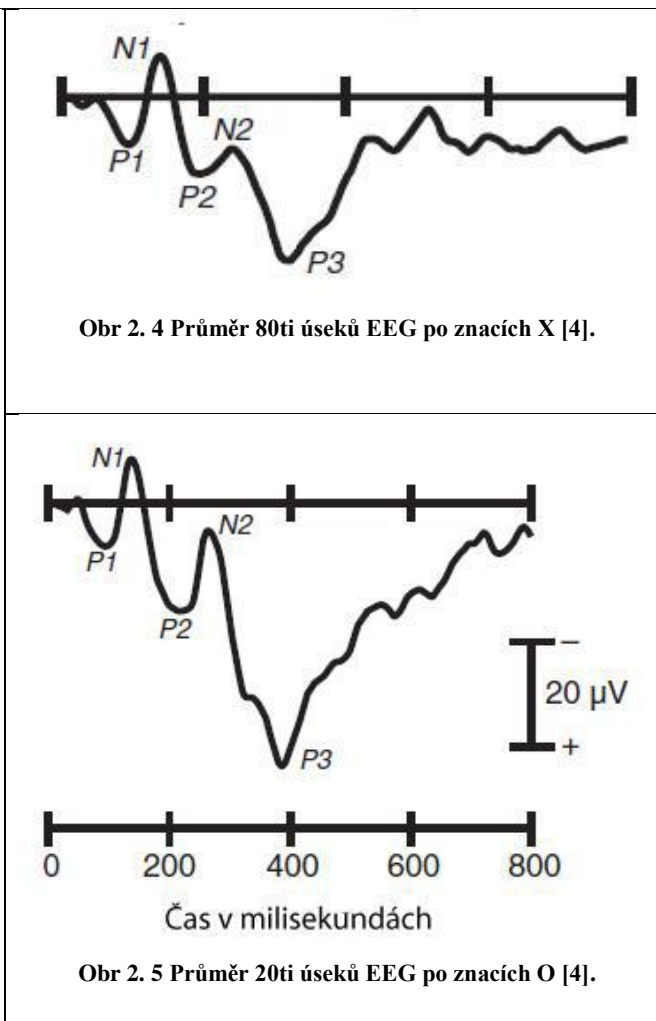
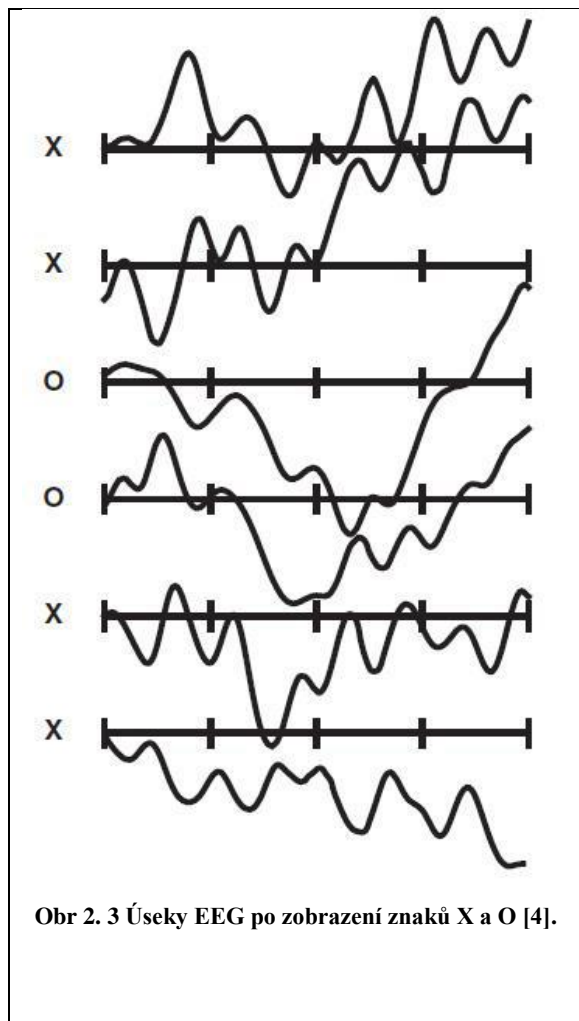


Obr 2. 2 Průběh EEG signálu při měření ERP experimentu [4].

„V ERP experimentech se používají EEG signály vztažené k bodům v čase, kdy testovaný subjekt byl vystaven působení stimulu podle připraveného testovacího scénáře. Stimulem označujeme nějaký děj, proces nebo činnost působící na některý ze smyslů člověka. Vyhodnocuje se velmi krátký úsek průběhu signálu, který nastal v důsledku působení stimulu a reflektuje vjem zaznamenaný mozkem. Jelikož signál může být ovlivněn i jinými činiteli (jako je rušení signálu nebo stimul, který nebyl naplánován ve scénáři), často se měření na tentýž stimul opakuje několikrát a EEG signály se následně průměrují.“ [6]

„Typickým příkladem ERP experimentu může být sledování mozkové aktivity při vnímání dvou typů stimulů, z nichž je jeden výrazně častější. Konkrétně na příkladu experimentu ze zdroje [4], kde subjekt sledoval na monitoru náhodně se zobrazující znaky X a O. Na obrázku 2.2 je vidět průběh EEG signálu na jedné z elektrod. Na obrázku jsou vyznačeny shodně dlouhé časové úseky, které byly naměřeny poté, co byl subjektu zobrazen stimul (znak X nebo O na obrazovce). V rámci ERP experimentu není potřeba sledovat signál v době, kdy nepřišel žádný stimul, a tak signál mimo vyznačené oblasti na obdélníku se dále nevyužívá.“ [6]

„Na obrázku 2.3 jsou vidět jednotlivé průběhy přehledně pod sebou. Signály při stejných stimulech vyznačují jistou podobnost. Aby se vyloučilo rušení a ovlivnění jinými stimuly, provádí se průměrování signálů pro stejné stimuly (viz Obr 2. 4 a Obr 2. 5).“ [6]



2.4 Vyhodnocení ERP experimentu

Na zprůměrovaných úsecích, které následovaly po stejných stimulech, se vyhodnocují lokální maxima, která se označují jako vlny (v angličtině peak). Vlny se označují písmeny P a N, které jsou následované číslem, jak je tomu na obrázku 2.4. P označuje pozitivní vlnu a N negativní. Číslo pak označuje jejich pořadí od počátku úseku. Často jsou vlny označovány jako P100, N100, P200, atd. Číslo je v milisekundách a určuje název vlny podle času, ve kterém nastala. Ovšem toto označení vychází z prvních měření, kdy shodou okolností vycházela vlna P3 300ms po stimulu, a tak označením P300 se má na mysli třetí pozitivní vlna, i když může nastat i více než 400ms po stimulu ($P3 = P300$) [4]. Z experimentu vyplývá, že P3 vlna je daleko větší u méně frekventovaného znaku O, než u hodně frekventovaného znaku X. Tato vlna je z hlediska ERP nejzajímavější a nejprozkoumanější vlnou, která má nejen široký experimentální význam, ale také stále větší význam klinický. [5]

Vlna P3 je potenciálový komplex pozitivních a negativních vln s dominantní pozitivitou o latenci cca 300ms od okamžiku stimulace. Bývá označována za neurofyzilogický korelát ukončení procesu správné detekce známého podnětu, kterému je přisuzován určitý význam. P300 je endogenní odpověď, nezávislá na modalitě použitého podnětu. [5]

3 Software pro prezentaci stimulů

„Při měření evokovaného potenciálu se používají testovací scénáře, což jsou sady různých stimulů v různých kombinacích a načasováních, na základě kterých vznikají odezvy v mozku měřitelné EEG přístrojem. Tyto odezvy jsou cílem dalšího výzkumu. Navržením vhodného scénáře je možné zkoumat např. účinky podprahového vnímání, dobu mezi zaregistrováním stimulu mozkiem a fyzickou reakcí subjektu atd. Jelikož reakce mozku následuje velmi krátce po zobrazení stimulu, používá se k zobrazování stimulů specializovaných programů, které umožňují synchronizaci s elektroencefalografem. Programy na prezentaci stimulů umožňují vytváření testovacího scénáře a jeho následnou prezentaci testovanému objektu. Program musí umět komunikovat s elektroencefalografem, aby mohl synchronizovat počátek zobrazení stimulu s časem v elektroencefalografu. Zásadní je v tomto případě minimální doba odezvy. Dlouhá nebo proměnlivá latence znehodnocuje naměřené výsledky.“ [6]

3.1 Seznam softwarů pro prezentaci stimulů

Pro srovnání s PreSti byly vybrány různé komerční placené softwary. Dále bude popisována jejich funkčnost, způsob vytváření testovacích scénářů, licence, výhody a další parametry. Na závěr budou tyto aplikace vzájemně porovnány.

3.1.1 DirectRT

Program byl vytvořen pro poznávací a kognitivní úlohy, které požadují rychlou odezvu a prezentaci stimulů. Aplikace DirectRT je vytvořena firmou Empiresoft [8] a je určená pro platformu Windows od verze XP výš. Využívá knihovnu DirectX.

V DirectRT lze vytvářet testovací scénáře prostřednictvím jednoduchých stimulů a to bez nutnosti znalosti programování. Testovací scénáře mohou obsahovat stimuly typu obrázek, animace, zvuk. K vytvoření scénáře je nutné si vytvořit input soubor (viz Obr 3. 1), který je ve formátu tabulky (soubor s příponou csv). Je nutné nadefinovat sloupce, které reprezentují vlastnosti stimulů. Řádky reprezentují jednotlivé stimuly. Program při spuštění stimulu začne ukládat výstupní data do output souboru. Mezi tato data patří například čas odezvy uživatele na stimuly, odpovědi na jednotlivé stimuly atd. Program efektivním způsobem sbírá a třídí tato data. Umožňuje lehce zaznamenávat a oddělovat informace o jednotlivých subjektech a podmínkách experimentu.

Vytváření scénářů

Scénář se vytváří v souboru s příponou csv. Je to formát souboru, který je určen k tvorbě tabulek (viz Obr 3. 1). Jednotlivé hodnoty jsou odděleny standardně znakem čárky nebo středníkem. Toto nastavení je závislé na jazykovém nastavení tabulkového editoru, ve kterém uživatel pracuje. Pro bezproblémové spuštění scénáře z csv souboru je nezbytné zachovat jeho správný formát, ve kterém jsou jednotlivá data oddělena čárkami. Standardně nový csv soubor obsahuje tabulku s 11 sloupci. Sloupce představují nastavitelné atributy, které slouží k nadefinování stimulů podle představ uživatele. Jednotlivé řádky reprezentují konkrétní stimuly se specifickými vlastnostmi a nastavením. Pro usnadnění a přehlednost vytváření stimulů aplikace používá dalších pomocných souborů, na které se lze z hlavního input souboru odkazovat.

block	trial	bgr	wgr	style	stim	loc	time	*
1	1	0	0	1	pic01	0,0,1	rt:any	*
1	2	0	0	1	pic02	0,0,1	rt:any	*
1	3	0	0	1	pic03	0,0,1	rt:any	*
1	4	0	0	1	pic04	0,0,1	rt:any	*
1	5	0	0	1	pic05	0,0,1	rt:any	*
1	6	0	0	1	pic06	0,0,1	rt:any	*
1	7	0	0	1	pic07	0,0,1	rt:any	*
1	8	0	0	1	pic08	0,0,1	rt:any	*
1	9	0	0	1	pic09	0,0,1	rt:any	*
1	10	0	0	1	pic10	0,0,1	rt:any	*

Obr 3. 1 Input soubor aplikace DirectRT (testovací scénář převzat z tutoriálu) [8].

Funkce

Všechny stimuly ve scénáři je možné libovolně načasovat. Stimuly mohou být typu text, obrázek, zvuk. Obrázky je možné zobrazovat náhodně z určité množiny. DirectRT umožňuje přehrávání zvukových stimulů a také jejich nahrávání. Je možné definovat jednotlivé klávesy, které slouží k zaznamenání odpovědi subjektu na daný stimul.

Komunikace s externím zařízením

DirectRT pro komunikaci mezi počítačem a externím zařízením používá TTL signály. Data přes TTL je možné zasílat a přijímat za předpokladu, že počítač i externí zařízení vlastní buď sériové či paralelní porty.

Licence

Uvedené ceny jsou bez daní. Cena licence se pohybuje okolo 500,- USD.

- Basic – základní licence, vyplatí se pro malé množství PC.
- Small group – skupinová licence pro maximálně 25 PC.

- Departmental site – omezená na oddělení (aktivace pomocí IP).
- Campus – neomezená licence pro celou společnost.

Odezva

Společnost Empiresoftware garantuje časovou odezvu v jednotkách milisekund pro zpracování obrázků, zvuku, prezentaci výstupu a komunikaci s externím zařízením.

Výhody

DirectRT je vhodný pro experimenty, které nevyžadují vytváření složitých scénářů. Vytváření scénářů je jednoduché a intuitivní. Je vhodný i pro uživatele, kteří nemají žádné zkušenosti s programováním. Postup vývoje scénáře je vždy stejný. Bohužel tato vlastnost je i nevýhodou, protože vytváření stimulů je velice omezené tím způsobem, že pro nestandardní scénáře se stává víceméně nepoužitelným.

3.1.2 Presentation

Rozšířený experimentální software od firmy Neurobehavioral Systems [9]. Presentation je aplikace určená pro prezentaci stimulů v neuro experimentech. Program je určený pouze pro platformu Windows. Vytváření testovacích scénářů se provádí v editoru, který Presentation obsahuje. Testovací scénáře se vytváří ve vlastním programovacím jazyce. Presentation má podporu pro zasílání vizuálních (2D a 3D), zvukových a video stimulů. Disponuje možností je prezentovat v testovacím scénáři jednotlivě nebo ve volitelné kombinaci.

Vytváření scénářů

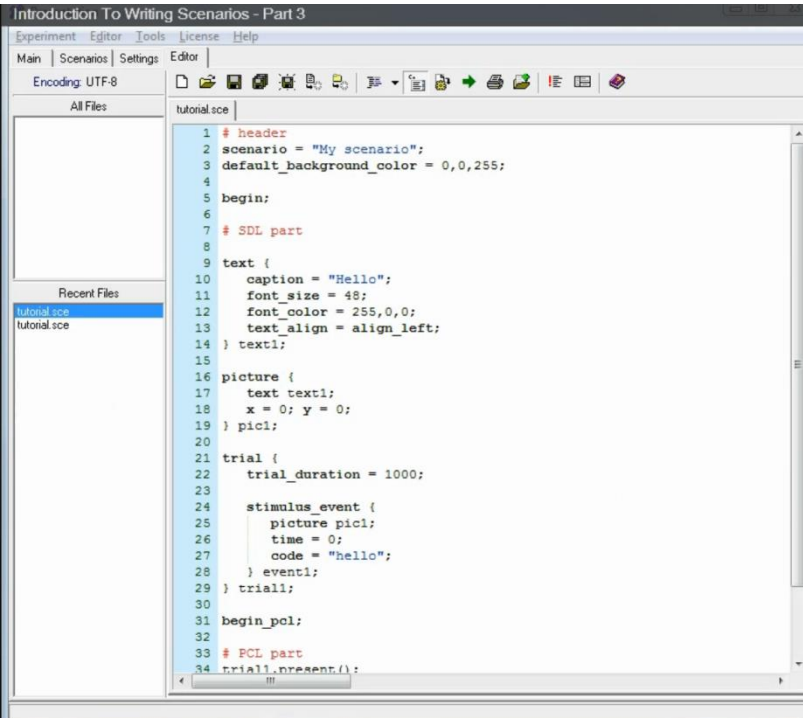
Testovací scénář se vytváří ve vlastním, strukturovaném a skriptovacím programovacím jazyce (viz Obr 3. 2). Scénář se vytváří v editoru, který je součástí programu. Tento editor dokáže zvýrazňovat a kontrolovat syntaxi zdrojového kódu. Scénář se vytváří s pomocí 2 jazyků:

- SDL (Scenario Description Language) – jazyk určený k popisu stimulů a specifikování jejich vlastností.
- PCL (Presentation Control Language) – interpretovaný programovací jazyk používaný k implementaci průběhu scénářů.

Testovací scénář se dělí na 3 části:

1. Header – začátek testovacího scénáře, kde se definují parametry, které mají vliv na celý scénář, například jméno scénáře (zobrazeno ve stavovém okně, když se scénář spustí).
2. SDL – definice všech stimulů, jejich komponent a posloupností stimulů, které budou použity ve scénáři (deklarace, definice a inicializace objektů).
3. PCL – zde se implementuje průběh scénáře a určuje se zde chování jednotlivých objektů nadefinovaných v SDL.

Presentation v průběhu scénáře vytvoří výstupní soubor Logfile. Ten obsahuje informace o událostech, které nastaly během scénáře. Může také obsahovat tabulku událostí, stimulů, videí a další užitečné informace.



```

1 # header
2 scenario = "My scenario";
3 default_background_color = 0,0,255;
4
5 begin;
6
7 # SDL part
8
9 text {
10  caption = "Hello";
11  font_size = 48;
12  font_color = 255,0,0;
13  text_align = align_left;
14 } text1;
15
16 picture {
17  text text1;
18  x = 0; y = 0;
19 } pic1;
20
21 trial {
22  trial_duration = 1000;
23
24  stimulus_event {
25    picture pic1;
26    time = 0;
27    code = "hello";
28  } event1;
29 } trial1;
30
31 begin_pel;
32
33 # FCL part
34 trial1.present();

```

Obr 3. 2 Presentation – snímek obrazovky (testovací scénář převzat z tutoriálu) [9].

Analyzér

1. Slouží k rychlé zpětné vazbě. Po skončení spuštěného scénáře zobrazuje ve vývojovém editoru data, která se ukládají do logfile souboru.
2. Slouží ke klasifikaci událostí a úpravě dat do výchozích formátů. Tyto úpravy se dají použít jako příprava pro pozdější použití analytických nástrojů. Analyzér poskytuje možnost naprogramovat si svůj vlastní filtr a vytvořit si nastavení tisku položek dat do výchozího souboru.

Funkce

V Presentation je možné vytvářet 2D a 3D vizuální, zvukové a video stimuly, které je možné prezentovat i ve stejném čase. K vytváření testovacích scénářů se používá programovací jazyk, který mimo jiné také umožňuje vytvořit vlastní uživatelské rozhraní scénáře. Program poskytuje implementované rozhraní určené pro připojení externího zařízení. Pro komfortní programování je součástí aplikace editor kódu, zvýrazňující syntaxi a doplňující psaný kód. Presentation také poskytuje rozhraní pro programovací jazyk Python, které umožňuje vytváření testovacích scénářů v tomto jazyce. Aplikace dále obsahuje funkčnost pro měření času výskytu událostí a jeho kontrolu. Aplikaci je možné propojit s Matlabem. Presentation je rozšiřitelný vytvářením programovatelných zásuvných modulů.

Komunikace s externím zařízením

Propojení s externím zařízením je zařízeno komunikací TTL signály přes sériový a paralelní port.

Licence

Cena standartní licence se pohybuje okolo 500,- USD. Zakoupená licence se vztahuje na všechny verze Presentation.

- Typy licencí
 - Code – aktivace produktu přes webové stránky.
 - Key – používá USB zařízení, ve kterém je uložený klíč. Když chce uživatel spustit Presentation, tak mu stačí vložit flashdisk s licencí do počítače s nainstalovanou aplikací. Výhoda této licence je její přenositelnost na jakýkoli počítač.
 - Web – umožňuje spouštět experimenty na neomezeném počtu počítačů. Nutnou podmínkou pro používání této licence je připojení počítače k internetu ve chvíli, kdy chce uživatel s experimentem pracovat. Jedna licence je vázáná na jeden zaregistrovaný experiment. Je možné kdykoli a neomezeně měnit experiment, který se vztahuje k webové aktivaci.

- 1. Standard – licence může být zakoupena kýmkoliv, kdo nespádá do kategorie student a teaching.
- 2. Student – studenti si mohou zakoupit licenci za sníženou cenu s použitím jednoho aktivačního klíče a dobou platnosti 1 rok.
- 3. Teaching – licence pro vyučování v učebně, za každý přidaný počítač jednotná sazba s platností na jeden rok.

Výhody

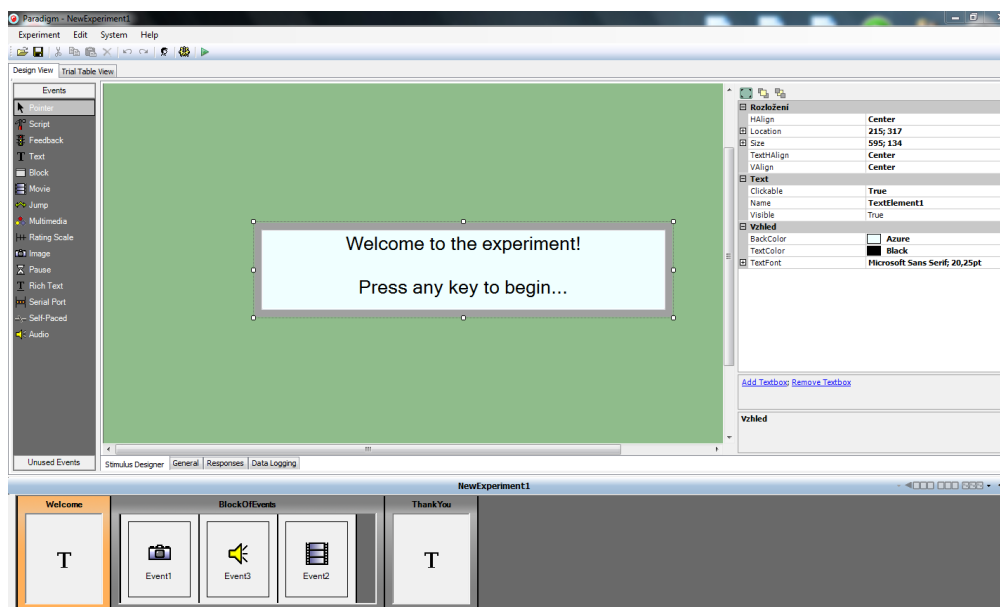
Uživatel používající Presentation má volnost při vytváření testovacích scénářů, protože scénář je možné naprogramovat. Programovatelnost výrazně rozšiřuje možnosti vytváření scénářů. Tato skutečnost má i svou nevýhodu, a to náročnost na uživatele, který musí umět programovat, což velice stěžuje využitelnost tohoto softwaru v oblasti medicíny. Lze naprogramovat rozsáhlý testovací scénář, ale v případě potřeby velice jednoduchého scénáře je nutné tento scénář také naprogramovat.

3.1.3 Paradigm

Paradigm [10] je software určený k vytváření testovacích scénářů a jejich prezentaci. Je určený pro platformu Windows a využívá knihovnu DirectX. Testovací scénáře se vytváří v grafickém editoru s využitím technologie drag and drop. Většinu scénářů lze vytvořit bez nutnosti programování. Pro kompletní flexibilitu Paradigm poskytuje rozhraní pro jazyk Python, které umožňuje rozšířit základní funkce o psaní skriptů v tomto jazyce.

Vytváření scénářů

Testovací scénář v Paradigm se vytváří přidáváním takzvaných událostí do bloků pomocí technologie drag and drop (viz Obr 3. 3). Bloky slouží k organizaci událostí a vkládají se do posloupnosti. Pořadí jednotlivých bloků je ekvivalentní pořadí, ve kterém budou jednotlivé stimuly prezentovány. Události reprezentují stimuly, skoky, pauzy, zdrojové kódy atd. Každá událost v Paradigm má svoje vlastnosti, ve kterých se dají nastavit parametry, například časové rozmezí prezentace stimulu, jméno stimulu atd. Všechny události, které reprezentují vizuální stimuly, mají k dispozici designer, nastavující pozici textu, barvu, písmo atd. K dispozici je také dialogové okno, ve kterém je možné nastavit vstupní zařízení pro poskytování odpovědi uživateli. Toto nastavení lze použít například v situaci, kdy experiment požaduje nastavení dvou kláves pro kladnou a zápornou odpověď testovacího subjektu. Ke každé události je možné vytvořit seznam zdrojových dat. Například v případě vytváření obrázkových stimulů se k události připojí seznam se zdroji obrázků, kde je také možné nastavit, jakým způsobem se budou obrázkové stimuly prezentovat, zda sekvenčně, náhodně nebo jiným způsobem.



Obr 3. 3 Paradigm - snímek obrazovky [10].

Funkce

Paradigm je software určený pro vytváření a prezentování vizuálních a zvukových stimulů. Vizuální stimuly se dělí na stimuly typu text, obrázek a video. Stimuly lze prezentovat jednotlivě nebo současně, a to v náhodném či uživatelem zvoleném pořadí. Aplikace umožňuje definovat a nastavovat externí vstupní zařízení, které poskytuje uživateli schopnost reagovat na stimuly. Aplikace umožňuje přiřadit stimulům v editoru událost, na kterou lze zareagovat. V Paradigm je možné nadefinovat data, která se při spuštění a provádění scénáře zaznamenají do výstupního souboru. Například odpovědi uživatele, chyby, čas začátku a konce stimulu atd. V případě, že uživateli nebudou stačit standardní funkce, existuje také možnost použít skriptovací jazyk Python, který mu poskytne větší expresivitu.

Vzdálená komunikace

Program Dropbox poskytuje distribuci a přístup k datům z jakéhokoli místa, které má připojení k internetu. V kombinaci s Paradigm přehrávačem je možné spouštět a upravovat testovací scénáře téměř odkudkoli. Pro samotné spouštění scénářů v přehrávači není nutné vlastnit licenci.

Komunikace s externím zařízením

Paradigm komunikuje s externím zařízením pomocí sériového a paralelního portu. Umožňuje komunikaci s více zařízeními současně. Například lze kombinovat zařízení EEG a eye tracker. Je také možné zakoupit USB – Parallel Port Converter, který umožní připojit externí zařízení s USB vstupem k paralelnímu portu počítače.

Licence

Licence se rozdělují na Standard a Student. Zakoupená licence se vztahuje k určité verzi a jedné kopie programu.

- Standard – cca 500,- (USD).
- Student – cca 150,- (USD).

Odezva

Pro prezentaci stimulů a komunikaci s externím zařízením poskytuje Paradigm standardně časovou odezvu s přesností na jednotky milisekund.

Výhody

Největší výhodou Paradigm je vytváření testovacích scénářů v grafickém rozhraní. Vytváření scénáře je velice rychlé, efektivní a nenáročné na čas. V porovnání s DirectRT je uživatel schopen vytvořit pokročilejší testovací scénář za velice krátkou dobu. Seznámení uživatele se softwarem není časově náročné. Flexibilita vytváření scénářů je zajištěna spoluprací Paradigm s programovacím a skriptovacím jazykem Python.

3.1.4 PreSti

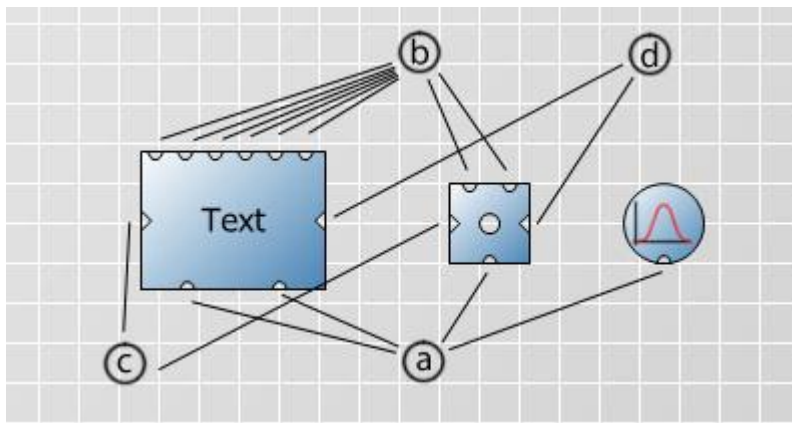
PreSti je svobodný, rozšiřitelný software určený k prezentaci stimulů a vytváření testovacích scénářů. PreSti je určený pro platformu Windows. K prezentování stimulů používá knihovnu DirectX. K vytváření testovacích scénářů využívá oproti jiným komerčním softwarům vizuální programování. Vizuální programování poskytuje výhodu rychlejšího a přehlednějšího způsobu vytváření testovacích scénářů než v případě klasického programování.

Vytváření scénářů

Základními prvky vizuálního programovacího jazyka PreSti jsou takzvané symboly. Symboly reprezentují stimuly nebo prvky pro řízení testovacího scénáře a další funkcionalitu. Symbolům je možno měnit jejich vlastnosti. Tyto vlastnosti například zahrnují dobu prezentace stimulu, cílovou cestu ke zdroji, pozici obrázku na obrazovce atd. Vlastnosti

symbolů jsou závislé na jejich typu. Všechny symboly kromě spojovacích obsahují porty. V PreSti existují 4 druhy portů (viz Obr 3. 4):

- a) Výstup hodnoty z objektu (Getter).
- b) Vstup hodnoty do objektu (Setter).
- c) Událost (Event).
- d) Akce (Action).



Obr 3. 4 Presti - rozmístění portů na symbolech [6].

Pro zajištění správné funkčnosti je nutné mít možnost symboly propojovat. K tomu slouží 2 typy spojení.

- Spojení (Connection)
 - „Používá se k propojení getterů a setterů a umožňuje tím přenášet hodnoty mezi objekty. Přenos hodnoty iniciuje vždy objekt s setterem na své straně spojení. V diagramu je spojení označeno plnou čarou.“ [6]
- Spuště (Trigger)
 - „Slouží k propojení akcí a událostí. Spuště určuje akci, kterou bude reagováno na událost. Nejběžnějším použitím spouště je reakce na událost ukončení stimulu spuštěním akce start jiného stimulu. Spuště má jedinou vlastnost – zpoždění, kterým lze určit prodlevu mezi vyvoláním události a vykonáním akce. V diagramu je spouště označena šipkou ve směru od události k obslužené akci.“ [6]

Funkce

V PreSti je možné vytvářet vizuální a zvukové stimuly. Vizuální se dělí na textové a obrázkové. Aplikace má k dispozici generátory náhodných čísel. Vygenerované hodnoty se dají použít jako vstupní hodnoty portů u jednotlivých symbolů. Pro odpovědi subjektu na jednotlivé stimuly PreSti poskytuje možnost subjektu reagovat pomocí myši a klávesnice. Program dokáže komunikovat s přístrojem na měření EEG signálů po paralelním portu. Pokud si návrhář při sestavování testovacího scénáře nevystačí se stávajícími symboly, je možno rozšiřovat PreSti pomocí modulů.

Komunikace s externím zařízením

Komunikace s externím zařízením je dostupná přes paralelní port.

Licence

Tento program je volný software. Používání programu je zdarma. Může se šířit a modifikovat podle ustanovení Obecné veřejné licence GNU, vydávané Free Software Foundation.

Odezva

Průměrná odezva mezi počátkem stimulu a zahájením komunikace přes paralelní port se standardně pohybuje do 3 milisekund. Maximální latence PreSti přesahuje hodnotu 20 milisekund. Tato latence se vyskytuje v max. 10 % případech. Aby ERP experimenty nebyly ovlivňovány latencí, je nutné se dostat pod hodnotu 5 ms.

Výhody

V dnešní době zatím neexistuje známý a dostupný software určený pro prezentaci stimulů, který by využíval při vytváření testovacích scénářů vizuálního programování. Vizuální programování urychluje vytváření testovacího scénáře a při správném návrhu vizuálního programovacího jazyka předchází syntaktickým chybám, ke kterým často dochází v textovém programování. Nevýhoda softwarů, které využívají k vytváření testovacích scénářů grafické rozhraní, je jejich nerozšiřitelnost. Uživatel má k dispozici konečnou množinu funkčních možností. V PreSti je rozšiřitelnost zajištěna možností vytvářet další funkční symboly.

3.2 Porovnání softwarů

Název softwaru	Platforma	Dostupnost (přibližně v USD)	Způsob vytváření testovacího scénáře
DirectRT	Windows	Licence – 500	Úprava csv souboru
Presentation	Windows	Licence – 500	Programování
Paradigm	Windows	Licence - 500	Grafické rozhraní – drag and drop
PreSti	Windows	free	Vizuální programování

Tabulka 3. 1 Porovnání softwarů

3.2.1 Funkcionalita

- DirectRT
 - Vizuální stimuly (obrázek, text).
 - Zvukové stimuly (přehrávání zvuků, nahrávání zvuků).
 - Náhodný generátor pořadí stimulů.
 - Možnost definovat zařízení pro odpovědi uživatele.
 - Chybí flexibilita při vytváření testovacích scénářů.
- Presentation
 - Vizuální stimuly (2D a 3D(obrázek, text), video, implementovatelné grafické rozhraní).

- Zvukové stimuly (přehrávání, nahrávání zvuků).
- Možnost definovat zařízení pro odpovědi uživatele.
- Flexibilita při vytváření testovacích scénářů zajištěna programováním v jazyce Python.
- Paradigm
 - Vizuelní stimuly (obrázek, text, video).
 - Zvukové stimuly (přehrávání zvuků, nahrávání zvuků).
 - Náhodný generátor pořadí stimulů.
 - Možnost definovat zařízení pro odpovědi uživatele.
 - Flexibilita při vytváření testovacích scénářů zajištěna možností vytvoření skriptu v jazyce Python.
- PreSti
 - Vizuelní stimuly (obrázek, text).
 - Zvukové stimuly (přehrávání zvuků).
 - Náhodný generátor čísel.
 - Možnost definovat zařízení pro odpovědi uživatele.
 - Flexibilita při vytváření testovacích scénářů omezená množinou symbolů vizuelního programovacího jazyka (možnost rozšířit).

Hlavní kritérium výběru softwaru pro prezentaci stimulů v oblasti medicíny je jednoduchost a rychlost vytvoření testovacího scénáře. Nejvíce intuitivní a nejrychlejší způsob vytvoření testovacího scénáře nabízí ve svém grafickém prostředí aplikace Paradigm. Použití grafického prostředí je také určitým způsobem limitující. V situaci, kdy nastane potřeba použít pokročilejší funkčnost, kterou grafické prostředí nenabízí, je nutné tuto funkčnost naprogramovat v jazyce Python. PreSti nabízí vytváření testovacích scénářů ve vizuelním programovacím jazyce, který odstraňuje tuto nevýhodu a to tak, že funkčnost si vytváří s pomocí symbolů sám uživatel. Funkčnost PreSti je omezena pouze množinou symbolů, která je rozšiřitelná. Porovnání softwarů (viz Tabulka 3. 1).

4 Specifikace požadavků

4.1 Úvod

Tento dokument slouží ke specifikaci požadavků, které budou použity k vylepšení funkčnosti softwaru PreSti v rámci bakalářské práce. Tato vylepšení posunou PreSti o další krok k jeho praktickému využití v praxi.

4.1.1 Rozsah práce

Tato práce bude navazovat na diplomové práce Petra Jaroše [6], který PreSti vytvořil a Petra Háky [7], který rozšířil jeho funkčnost. Software PreSti bude rozšířen o průvodce vytváření testovacích scénářů. Průvodce bude určen pro začínající i pokročilejší uživatele. Rozšíření funkčnosti poskytne těmto uživatelům vytvoření jednoduchých základních testovacích scénářů. Tyto scénáře průvodce vytvoří automatickým generováním grafů. Grafy budou pro uživatele k dispozici a začátečníkům též poslouží jako nástroj pro lepší pochopení fungování vizuálního programovacího jazyka.

4.2 Obecný popis

4.2.1 Kontext produktu

Aplikace pro svou správnou funkčnost vyžaduje:

1. Spuštění pod operačním systémem Windows verze XP a vyšší.
2. Nainstalovaný Framework .Net od verze 4.0.
3. Dostupnou knihovnu DirectX od verze 8.0.
4. Nainstalovaný Microsoft XNA Framework Redistributable od 4.0. [7]

4.2.2 Funkce produktu

1. Poskytnutí prostředku, který ulehčí vytváření základních testovacích scénářů začínajícími uživateli.
2. Možnost vytvoření hrubého návrhu testovacího scénáře, který bude sloužit jako základ pro pozdější úpravy a změny scénáře.

4.2.3 Třídy uživatelů

Produkt budou využívat 2 třídy uživatelů

- Začátečník – uživatel, který zatím nemá žádné zkušenosti s vytvářením testovacích scénářů v softwaru PreSti.
- Pokročilý – uživatel, který již disponuje základními zkušenostmi se softwarem. Je schopný sám navrhnout a vytvořit testovací scénář.

4.2.4 Omezení

Pro plnou funkčnost PreSti je nutné, aby počítač, na kterém bude software spuštěn, obsahoval paralelní port. Komunikace s elektroencefalografem vyžaduje spuštění PreSti s administrátorskými právy. [6]

4.3 Specifikační požadavky

Tato část se zabývá už pouze částí dodatečného rozšíření PreSti v rámci této bakalářské práce.

4.3.1 Funkce

Průvodce pro vytváření testovacích scénářů (začínající uživatelé)

Průvodce bude schopný pomoci uživateli s vytvářením jednoduchých základních testovacích scénářů. Toto rozšíření poskytne začínajícímu uživateli možnost vytvořit funkční testovací scénář bez předchozích znalostí a zkušeností práce v PreSti, tzn. bez nutnosti návrhu scénáře ve vizuálním programovacím jazyce.

Průvodce pro úpravu testovacích scénářů (pokročilejší uživatelé)

Jelikož je pokročilejší uživatel již schopný si sám navrhnout a vytvořit testovací scénář, je také schopný později upravit graf, který je generovatelný průvodcem. Pro tohoto uživatele bude možné využít průvodce jako nástroje k vytvoření hrubého konceptu grafu. Vytvořený koncept bude možné dále dle požadavků upravovat.

4.3.2 Požadavky na údržbu

Komentáře

Implementované rozšíření softwaru bude pro lepší pochopení jeho funkčnosti řádně okomentováno.

Verzování

Programové rozšíření aplikace bude, stejně jako jeho předchozí verze, postupně ukládáno na příslušném, zadavatelem definovaném úložišti s podporou verzování zdrojového kódu.

4.3.3 Licenční podmínky

Licenční podmínky jsou shodné s podmínkami původní verze PreSti (více informací [6]).

4.3.4 Autorská práva

Windows, .Net, DirectX, Microsoft XNA jsou registrované ochranné známky.

5 Návrh a implementace

5.1 Vizualní programovací jazyk

Pro vytváření testovacích scénářů PreSti používá vizualní programovací jazyk.

5.1.1 Definice vizualního programovacího jazyka

Vizualní programovací jazyk může pracovat s vizualními informacemi, podporovat vizualní interakci nebo umožnit programovat vizualním vyjádřením. Pouze poslední z definic se považuje za definici vizualního programování. Vizualní programovací jazyky se dále dělí podle typu a rozsahu vizualního vyjádření do jazyků založených na práci s ikonami, formuláři a diagramy. Vizualní programovací prostředí umožňují grafickým prvkům nebo ikonám, se kterými může uživatel interaktivně manipulovat, vytvářet programy podle jistých specifických gramatických pravidel. [3]

5.2 Základní testovací scénář

Za základní testovací scénář se dá považovat scénář, který je založen na principu zobrazování takzvaných target a nontarget stimulů [4]. Při zahájení experimentu se začnou s určitou pravděpodobností na obrazovce střídát různé zrakové vjemy, jako jsou například: písmena, číslice, obrázky atd. Následující testovací scénář je modifikací scénáře převzatého ze zdroje [4], kde target stimul reprezentuje písmeno Q a nontarget stimuly ostatní písmena abecedy. Pro náš případ budeme střídát různá slova, která budou reprezentovat barvy v anglickém jazyce. Target stimul je takový stimul, na který bude testovaný subjekt upozorněn ještě před začátkem experimentu. Testovací subjekt bude mít za úkol zaměřit se na výskyt target stimulu. Řekněme, že target stimul pro nás bude slovo: black s pravděpodobností zobrazení např. 40%. Množinu nontarget stimulů s celkovou pravděpodobností 60 % budou tvořit slova: purple, yellow, blue.

Při vytváření testovacího scénáře, který nebude založen na pravděpodobnostním výskytu stimulů jako testovací scénář v předchozím případě, je nutné zařídit, aby uživatel měl také možnost nějakým způsobem ovlivnit pořadí výskytu jednotlivých stimulů.

5.2.1 Upravitelnost testovacího scénáře vygenerovaného průvodcem

Z důvodu požadavků na funkcionalitu z pohledu pokročilejšího uživatele je nutné, aby bylo možné po vytvoření testovacího scénáře upravovat scénář, a tím splnit tento další požadavek.

5.2.2 Jednoduchost vytvoření testovacího scénáře pomocí průvodce

Začínající uživatelé, kteří ještě nemají s návrhem testovacích scénářů žádné zkušenosti, budou klást hlavní nároky na jednoduchost a intuitivnost uživatelského rozhraní. To by také mělo poskytovat návod či podrobný popis jednotlivých instrukcí, které povedou k úspěšnému vytvoření testovacího scénáře.

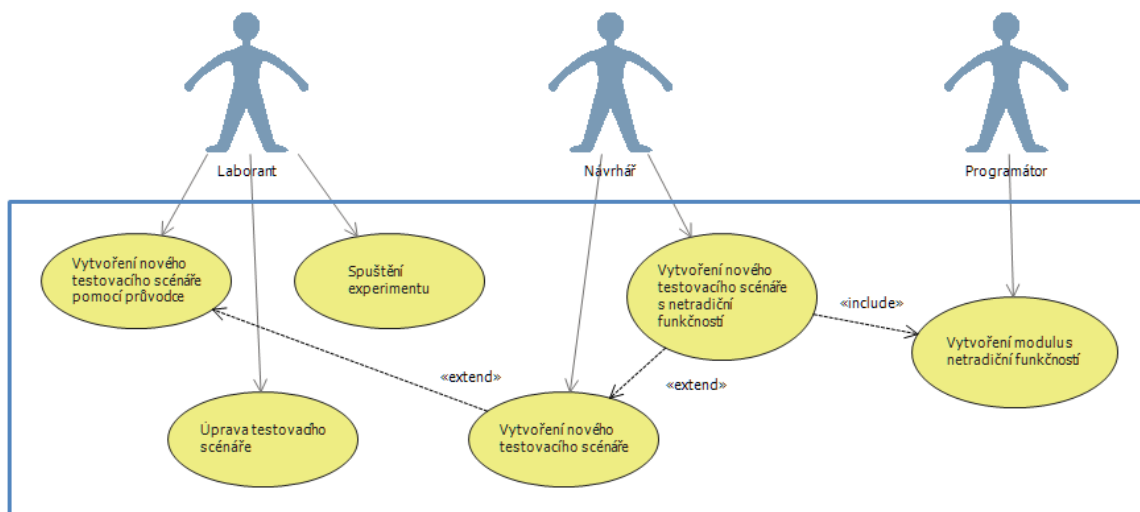
Požadavky na průvodce

- Možnost vytvářet stimuly, kterým bude možno nastavit pravděpodobnost jejich výskytu.
- Možnost ovlivnit pořadí výskytu stimulů.
- Možnost upravovat testovací scénář po vytvoření průvodcem.
- Intuitivnost, jednoduchost uživatelského rozhraní s možností nápovědy.

5.3 Kontext systému

UML Use Case diagram vytvořený Petrem Jarošem [6], zobrazující kontext systému, bude rozšířen o další případ užití (viz Obr 5. 1).

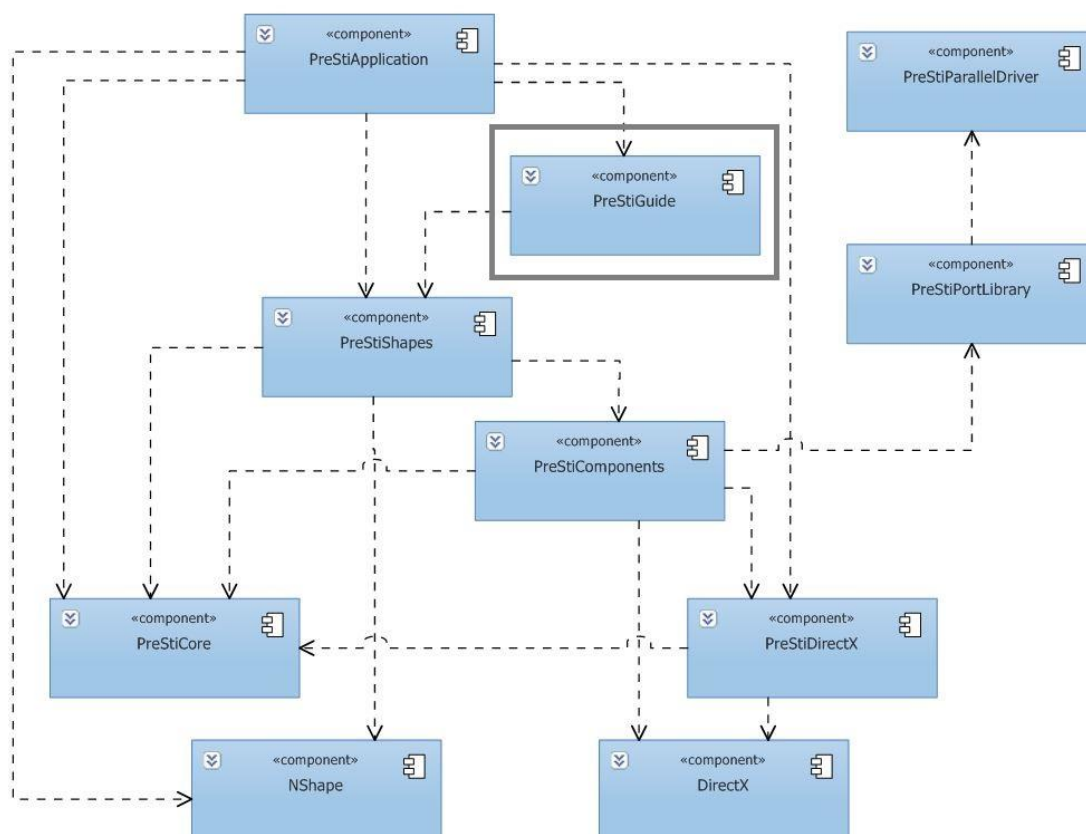
- Laborant
 - Uživatel, který provádí s programem ERP experimenty. Spouští pouze testovací scénáře vytvořené návrhářem. Případně provádí jednoduché změny parametrů testovacího scénáře. Po přidání nové funkčnosti bude schopen vytvářet průvodcem testovací scénáře. [6]
- Návrhář
 - „Uživatel programu schopný navrhovat testovací scénáře k měření evokovaného potenciálu pomocí jednoduchého grafického rozhraní.“ [6]
- Návrhář (programátor)
 - „Uživatel schopný navrhovat scénáře a navíc umí programovat v některém z podporovaných programovacích jazyků. Umí vytvořit nový modul do programu PreSti.“ [6]



Obr 5. 1 UML Use Case diagram rozšířený o další případ užití [6].

5.4 Architektura systému

PreSti je vyvíjeno pro platformu Windows, v jazyce C#, ve vývojovém prostředí Visual Studio 10. Aplikace je uložena na verzovacím serveru Codeplex¹. PreSti je rozdělena dle funkčnosti na více komponent, které jsou mezi sebou navzájem v interakci (viz Obr 5. 2). Každá komponenta v PreSti je řešena jako samostatný projekt ve vývojovém prostředí. Komponenty obsahují své vlastní třídy. Po překladu každá komponenta představuje samostatný binární soubor. Pro zachování řádné dekompozice je nutné, aby byla do balíku programu přidána další komponenta, která bude reprezentovat novou funkčnost (komponenta *PreStiGuide*).



Obr 5. 2 Diagram komponent programu PreSti rozšířený o PreStiGuide [6].

5.4.1 Původní sada komponent

PreStiCore

„Ústřední komponenta umožňuje vytvářet z vizuální reprezentace programu (navrženého testovacího scénáře) vnitřní reprezentaci objektů a následně program spustit. Obsahuje tedy interpret vizuálního programovacího jazyka. Dále obsahuje několik rozhraní a virtuálních tříd, které musí ostatní balíky implementovat, aby s nimi PreStiCore dokázal pracovat.“ [6]

¹Codeplex (<https://www.codeplex.com/>)

PreStiDirectX

„Komponenta zaobaluje funkcionalitu závislou na knihovně DirectX. Tato komponenta je oddělena od `PreStiCore` kvůli případnému snazšímu přechodu na jinou knihovnu.“ [6]

PreStiComponents

„Obsahuje třídy, které reprezentují funkcionalitu jednotlivých objektů ve vizuálním programovacím jazyce PreSti. Objekty mohou být jak stimuly testovacích scénářů, tak komponenty pro řízení běhu programu. `PreStiComponents` obsahuje základní sadu těchto objektů. Další je možné přidat jako rozšíření. Objekty neobsahují informace o vzhledu symbolu, který je bude reprezentovat. Vzhled symbolů je definován v komponentě `PreStiShapes`.“ [6]

PreStiShapes

„Je komponenta obsahující definice vzhledu symbolů. Je kompletně závislá na knihovně `NShape`. Z důvodu případné výměny knihovny `NShape` za jiný grafický framework pro vytváření diagramů je oddělena od komponenty `PreStiComponents`. Každý symbol je v `PreStiShapes` reprezentován třídou, ve které se definuje vektorově vzhled, pozice portů pro připojení a jiné vlastnosti symbolu.“ [6]

PreStiDriver

„Ovladač systému Windows umožňující komunikaci po paralelním portu.“ [6]

PreStiPortLibrary

„Knihovna napsaná v jazyce C++ umožňuje komunikaci mezi ovladačem a komponentou `PreStiComponents` psanou v C#.“ [6]

5.4.2 Rozšíření sady o komponentu `PreStiGuide`

Komponenta je schopna ulehčit vytváření testovacích scénářů začínajícím uživatelům a zároveň vytvářet hrubý, upravitelný koncept pro pokročilé uživatele. K těmto dvěma činnostem: vytváření a upravování testovacího scénáře, PreSti používá svůj vizuální programovací jazyk. Pro vytváření testovacích scénářů pomocí průvodce a pro editaci vygenerovaného grafu je nutná interakce této nové komponenty s vizuálním programovacím jazykem. Interakce je zajištěná závislostí `PreStiGuide` na komponentě poskytující symboly vizuálního programovacího jazyka `PreStiShape` (viz Obr 5. 2). Diagram zobrazuje další propojení `PreStiGuide` s jinou komponentou. Toto spojení reprezentuje závislost `PreStiApplication` na `PreStiGuide`. V tomto případě poskytuje své třídy prezentační vrstvě, kterou `PreStiApplication` obsahuje.

Typy informací zpracovávané komponentou PreStiGuide

Vstupní data

Vstupní data, která bude komponenta zpracovávat, budou poskytována vzájemnou interakcí mezi uživatelem a formulářem třídy `PreStiGuideWindow`. Mezi tato data patří například počet, typ stimulů, jejich atributy atd.

Výstupní data

Na základě výsledku zpracování vstupních dat komponenta vygeneruje výstupní graf, který se následně použije ke spuštění testovacího scénáře.

5.5 Implementace

Implementace nové funkcionality bude zachovávat koncept, který byl navržen Petrem Jarošem [6], rozdělení PreSti na komponenty a také důsledné oddělení aplikační vrstvy od prezentační.

5.5.1 Průvodce

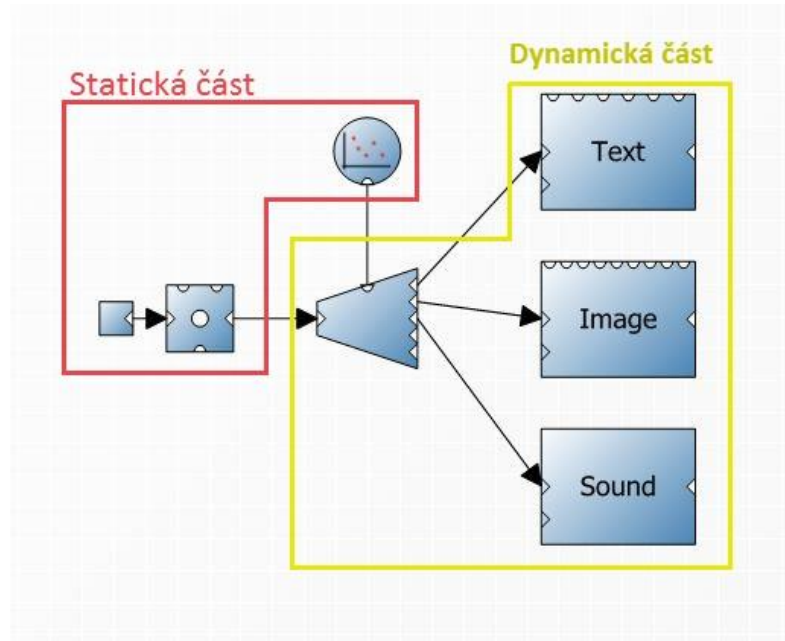
Je nutné implementovat takovou funkčnost, která umožní při vytváření stimulů nastavovat buď pravděpodobnost výskytu stimulů, nebo jejich pořadí. Vytváření testovacích scénářů je realizováno vytvářením grafů ze symbolů vizuálního programovacího jazyka. Funkčnost bude výhodné rozdělit do dvou tříd, a to z důvodu rozdílnosti grafů pro Pravděpodobnostního a Pozičního průvodce. Každá ze tříd bude definovat 2 metody, které vytvoří statickou a dynamickou část grafů (viz Obr 5. 3, Obr 5. 4).

- Statická část
 - symboly, které jsou vytvořeny a nastaveny automaticky při spuštění průvodce.
- Dynamická část
 - symboly, které jsou vytvořeny a nastaveny činnostmi uživatele.

Statická část bude obsahovat neměnný základ grafu. Ke statické části bude připojena dynamická část, kterou bude svou činností ovlivňovat uživatel aplikace. Některé metody tříd průvodce budou redundantní, protože kromě rozdílných grafů budou definovat částečně stejnou funkcionality. Z tohoto důvodu bude vytvořena rodičovská třída, od které budou předchozí dvě dědit.

Pravděpodobnostní průvodce

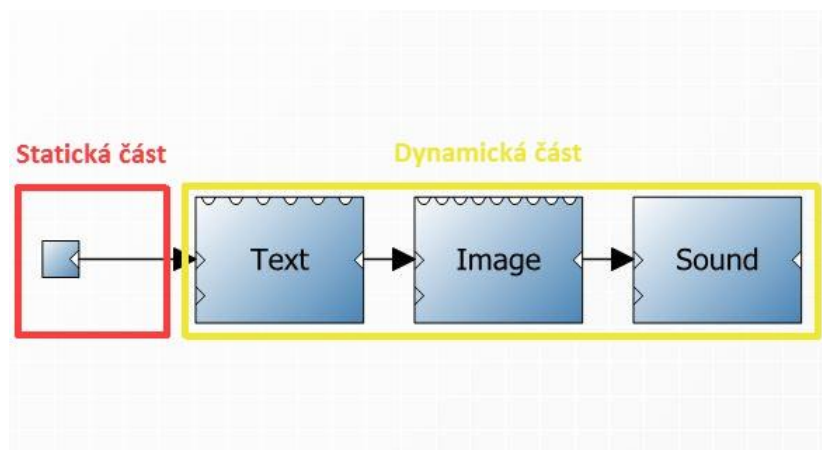
Průvodce bude poskytovat možnost určení pravděpodobnosti výskytu stimulů. Jeho funkčnost bude implementována třídou `RandomGuide`.



Obr 5.3 Graf Pravděpodobnostního průvodce.

Poziční průvodce

Průvodce bude poskytovat možnost určení pozice výskytu stimulů. Jeho funkčnost bude implementována třídou `SequentialGuide`.



Obr 5.4 Graf Pozičního průvodce.

5.5.2 Návrh tříd

Třídy v PreSti jsou děleny logicky do níže uvedených balíků. Podrobněji se zaměříme na balíky `cz.zcu.PreSti.Application` a `cz.zcu.PreSti.Guide`. V těchto balících bude implementována nová funkčnost.

Původní sada balíků

- `cz.zcu.PreSti`
 - „Hlavní balík reprezentující aplikační vrstvu programu.“ [6]
- `cz.zcu.PreSti.Shapes`
 - „Balík obsahuje jednotlivé symboly používané ve vizuálním programovacím jazyce PreSti. Obsahuje pouze informace o jejich vzhledu a chování při vytváření diagramů.“ [6]
- `cz.zcu.PreSti.Components`
 - „Balík objektů, které jsou vizuálně reprezentovány symboly z balíku `cz.zcu.PreSti.Shapes`.“ [6]
- `cz.zcu.PreSti.Application`
 - „Balík zaobaluje celou okenní aplikaci. Jedná se tedy čistě o třídy prezentační vrstvy.“ [6]

Rozšíření původní sady balíků

- `cz.zcu.PreSti.Guide`
 - Balík tříd, které reprezentují aplikační vrstvu průvodce.
- `cz.zcu.PreSti.Application`
 - Původní balík je rozšířen o nové třídy, které reprezentují prezentační vrstvu průvodce.

Popis tříd aplikační vrstvy

Vrstva je realizována balíkem `cz.zcu.PreSti.Guide` (viz Obr 5. 5). Skládá ze 6 tříd:

GuideParent

Abstraktní třída, která inicializuje a nastavuje členské proměnné předané z prezentační třídy `GuideWindow`. Ke své správné činnosti potřebuje třídy `SymbolsCreator`, `TemplatesShape` a `Property`. Třída je rodičem `SequentialGuide` a `RandomGuide`. Pro své potomky deklaruje 2 abstraktní metody: `AddDynamicGraphPart()`, `AddStaticGraphPart()`, které je nutné implementovat. Také poskytuje svým potomkům společné metody pro práci s vizuálním programovacím jazykem.

SequentialGuide

Potomek třídy `GuideParent` implementuje abstraktní metody svého rodiče a také využívá jeho další metody. Slouží k vytváření Pozičního průvodce `SequentialGuide` a definuje

metody, které umožňují vytvořit testovací scénář (graf vizuálního programovacího jazyka). Využívá třídy `GraphCreator`, která mu poskytuje potřebné symboly vizuálního programovacího jazyka. Díky parametrům, které dědí od svého rodiče, získává přístup k prezentaci dat a k interakci s formulářem třídy `GuideWindow`.

RandomGuide

Stejně jako `SequentialGuide`, je potomkem třídy `GuideParent`, implementuje abstraktní metody svého rodiče a používá jím poskytované metody. Slouží k vytváření Náhodného průvodce. Využívá třídy `GraphCreator`, která mu poskytuje potřebné symboly vizuálního programovacího jazyka. Díky parametrům, které dědí od svého rodiče, získává přístup k prezentaci dat a k interakci s formulářem třídy `GuideWindow`.

SymbolCreator

Vyváří a poskytuje symboly vizuálního programovacího jazyka třídě `ParentGuide` a jejím potomkům. Spolupracuje s třídou `TemplatesShape`, kterou využívá jako návratový typ svých metod.

TemplatesShape

Poskytuje třídě `SymbolCreator`, `GuideParent` a jejím potomkům nový datový typ. Obsahuje členské proměnné, které uchovávají informace o jednotlivých symbolech.

Property

Poskytuje třídě `GuideParent` a jejím potomkům nový datový typ, který v sobě uchovává objekty prezentační vrstvy (textbox, label).

Popis tříd prezentační vrstvy

Vrstva je realizována balíkem `cz.zcu.PreSti.Application` (viz Obr 5. 5). Do balíku byly přidány třídy `GuideMenuWindow`, `GuideWindow`. Skládá se ze 4 tříd:

GuideMenuWindow

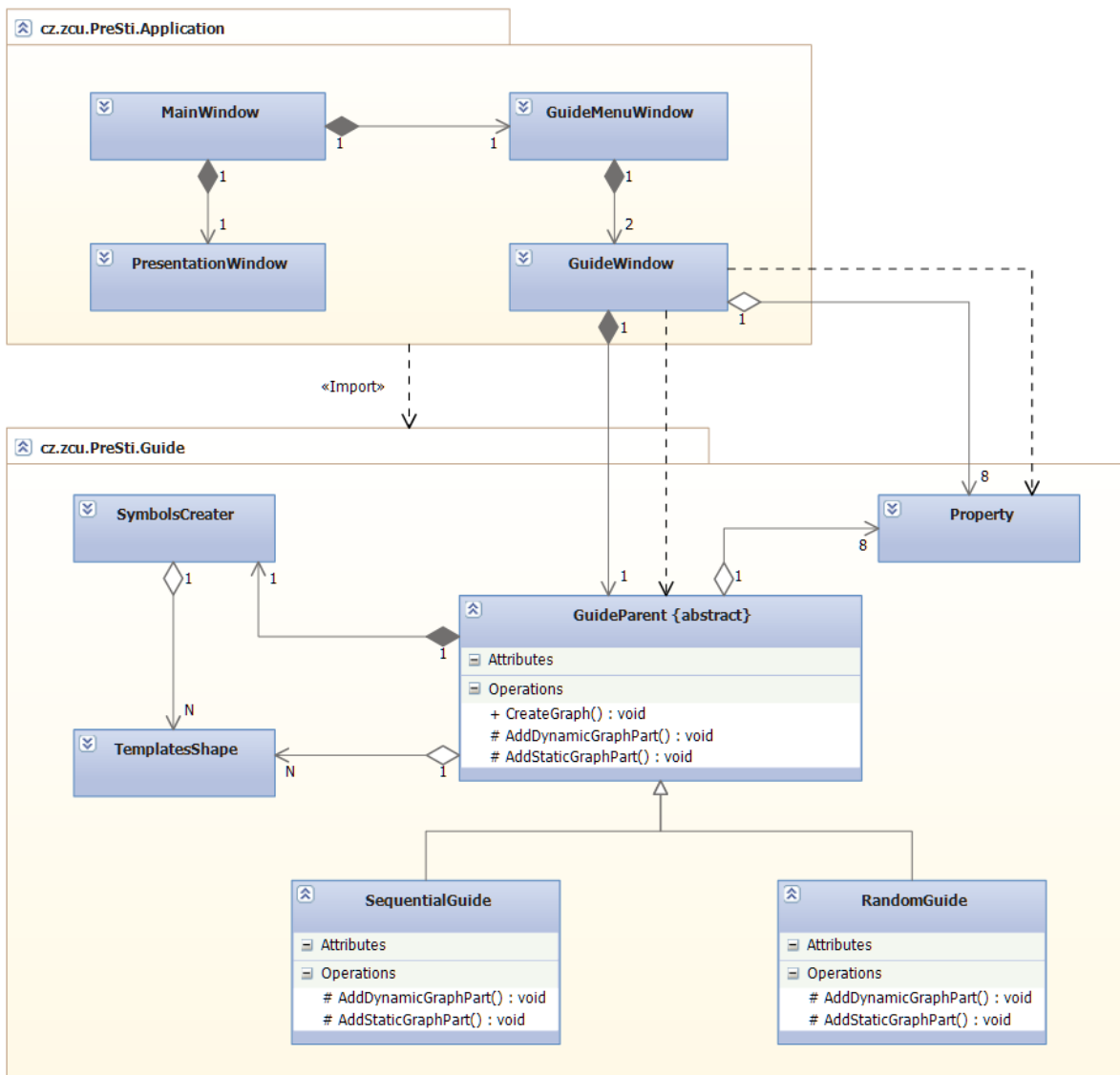
Reprezentuje úvodní formulář průvodce. Nabízí výběr z 2 typů testovacích scénářů.

GuideWindow

Třída, která definuje uživatelské rozhraní obou typů průvodců. Reprezentuje prezentační vrstvu, která spolupracuje s aplikační vrstvou `PreStiGuide`.

MainWindow

Formulář, který reprezentuje základní okno aplikace `PreSti`.



Obr 5. 5 Diagram tříd.

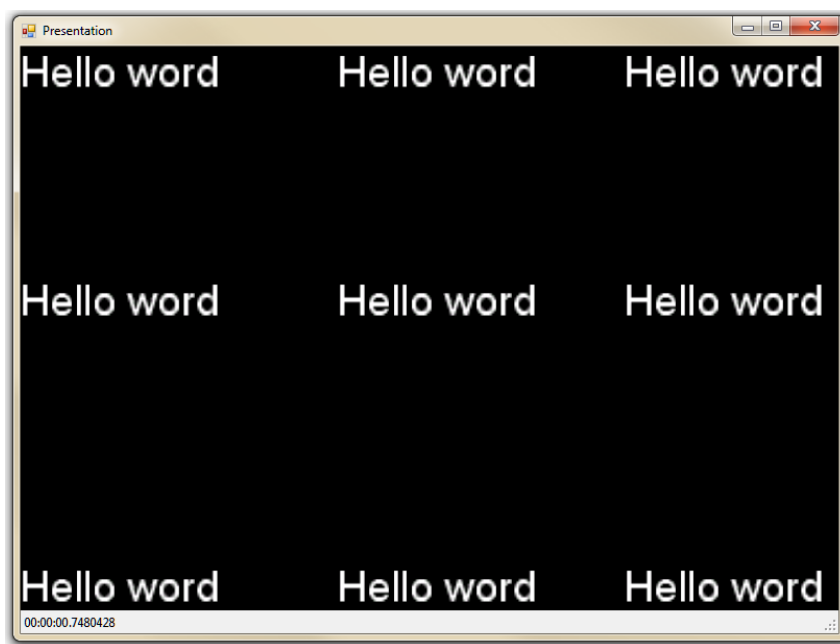
5.5.3 Dodatečné mimofunkční úpravy PreSti

Změna nastavení rozvržení okna aplikace.

Pro bezproblémové použití PreSti bylo nutné odstranit chybu v prezentační vrstvě. Chyba ovlivňovala správnou funkčnost hlavního okna. Zobrazování panelu dostupných symbolů při jeho rozšiřování bylo chybné, protože mělo vliv na správnou viditelnost vedlejšího panelu pro úpravu vlastností stimulů. Problém byl vyřešen dodatečnou úpravou layoutu.

Rozšíření funkčnosti symbolu textového stimulu

Na základě funkčních požadavků jednoduchosti a intuitivnosti vytváření testovacího scénáře byla přidána možnost pozicování textu symbolu textového stimulu. Původní funkcionality dovolovala pozicování zobrazovaného textu pouze zadáním horizontálních souřadnic X a vertikálních Y. Pro uživatele by bylo zdoluhavé vypočítávat souřadnice v případě požadavku na umístění textu například do středu prezentačního okna. Symbol byl rozšířen o vlastnosti `HorizontalAlignment` a `VerticalAlignment`. Kombinací těchto dvou typů vlastností je možno nastavit celkem 9 pozic (viz Obr 5. 6).



Obr 5. 6 Výčet možných pozic.

Přidáním nové vlastnosti symbolu vizuálního programovacího jazyka byla ztracena kompatibilita s testovacími scénáři z minulých verzí. Dostupné scénáře byly opět převedeny do formátu nynější verze, které jsou dostupné na CD v příloze bakalářské práce.

6 Testování

Přidaná funkčnost průvodce byla otestována vytvořením testovacích scénářů.

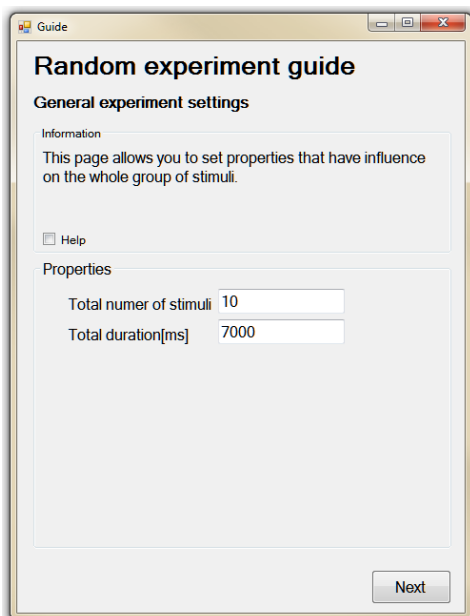
6.1 Pravděpodobnostní průvodce

Byl vytvořen základní testovací scénář, který je popsán v kapitole 5.2.

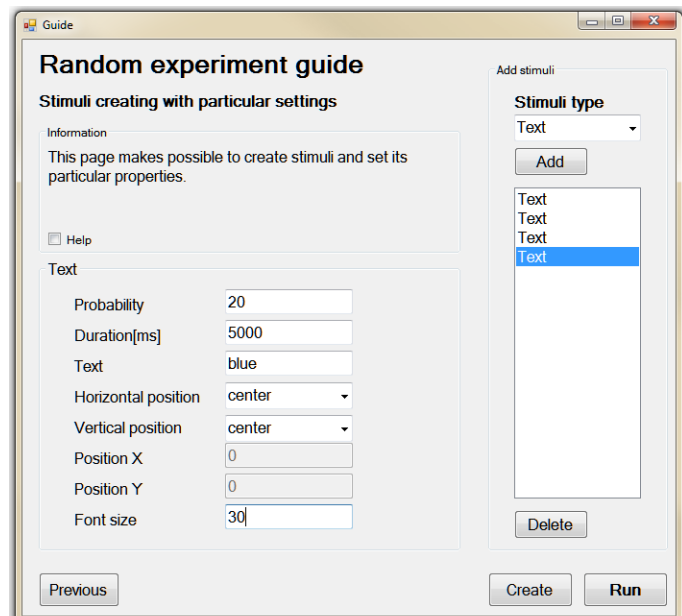
6.1.1 Vytvoření testovacího scénáře

Postup vytváření testovacího scénáře (viz Obr 6. 1, Obr 6. 2 a Obr 6. 3).

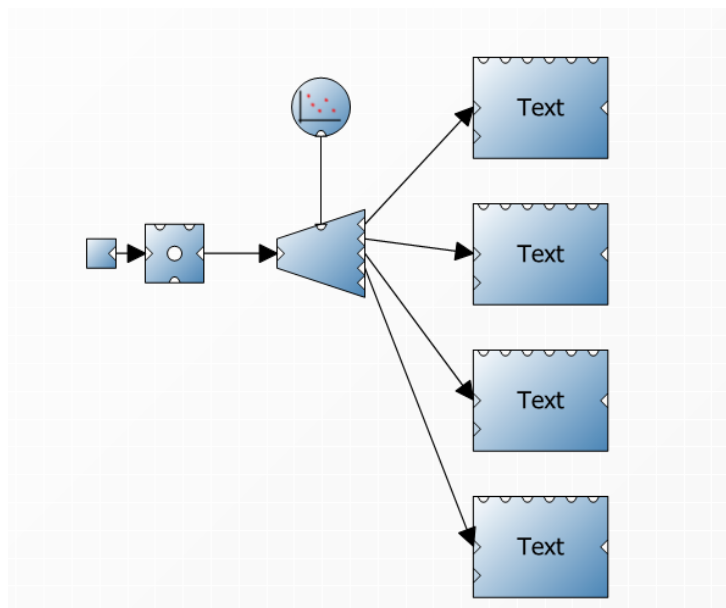
1. Nastavení globálních vlastností
 - Celkový počet zobrazených stimulů – 10.
 - Celková doba zobrazení jednoho stimulu, včetně interstimul intervalu – 7000 [ms].
 - Doba zobrazení jednoho stimulu – 5000 [ms].
2. Vložení textových stimulů
 - Target
 - Stimul1 – black (40%).
 - None-target
 - Stimul2 – purple (20%).
 - Stimul3 – yellow (20%).
 - Stimul4 – blue (20%).
3. Vytvoření grafu



Obr 6. 1 Nastavení globálních vlastností.



Obr 6. 2 Nastavení vlastností posledního textového stimulu.



Obr 6. 3 Graf vytvořený Pravděpodobnostním průvodcem.

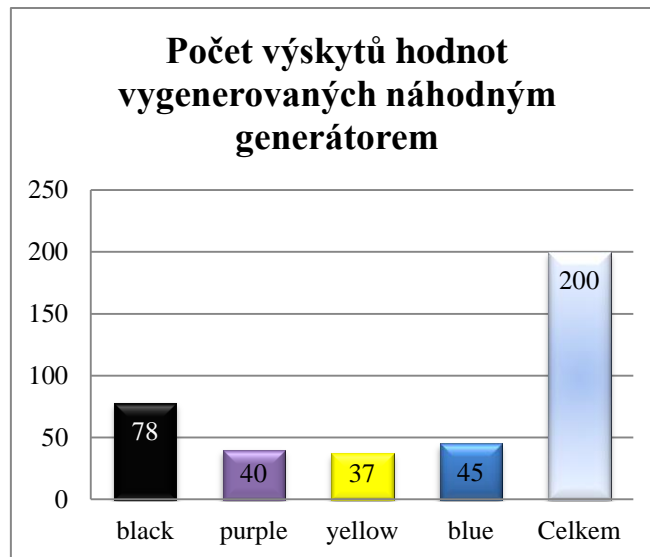
6.1.2 Zhodnocení funkčnosti testovacího scénáře

Průvodce vytváří spustitelný graf, který reprezentuje testovací scénář. Tento testovací scénář je plně funkční. Když je stimulů vytvořeno více než 4, vzniká problém. Ačkoli je graf průvodcem vygenerován správně, funkcionality spuštěného scénáře je chybná. Správná funkčnost scénáře je závislá na symbolu RandomGenerator. Symbol generuje náhodná čísla s určitou pravděpodobností. Vygenerovaná čísla jsou předávána symbolům typu Splitter pomocí spojovacího symbolu Connector. Connector zajišťuje přenos dat mezi symboly. Testovací scénář funguje správně za předpokladu, že je RandomGenerator propojen pouze s jedním Splitterem. V opačném případě dochází k problému. Problém vznikl již v původním návrhu PreSti, který s tímto případem větvení nepočítal. RandomGenerator by měl vygenerovat náhodnou hodnotu a poté ji v případě potřeby prostřednictvím Connectoru dalším objektům. Problém spočívá v tom, že hodnota je generována pro každé spojení Connectorem zvlášť. Z toho vyplývá, že není možné předat jednu stejnou vygenerovanou hodnotu z RandomGenerator do více Splitterů. Z tohoto důvodu je funkčnost testovacího scénáře omezena RandomGenerátorem pro maximální počet stimulů 4. Ačkoli je aplikace schopna vytvářet neomezený počet stimulů, uživatelé nedovolí vytvořit více než 4 stimuly, aby se vyhnula chybě.

Správná funkčnost generátoru

Pro ověření správné funkcionality testovacího scénáře vytvářeného Pravděpodobnostním průvodcem je nutné otestovat funkčnost generátoru náhodných čísel (viz graf na Obr 6. 4).

Pro potřeby testování byla hodnota celkového počtu zobrazených stimulů testovacího scénáře nastavena na číslo 200.



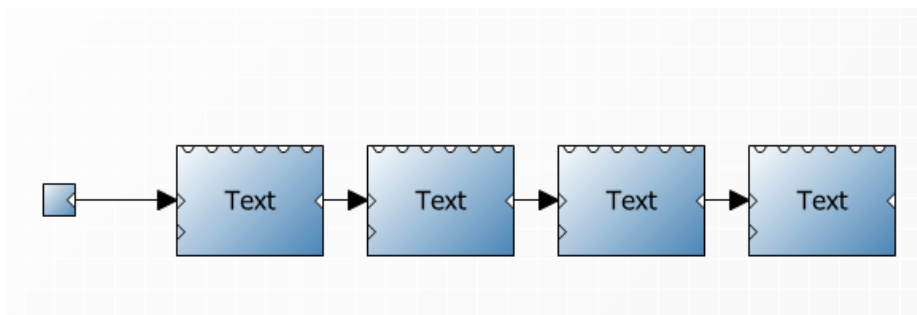
Obr 6. 4 Graf vygenerovaných hodnot.

Z grafu na Obr 6. 4 vyplývá, že generátor je schopný vygenerovat odpovídající hodnoty v závislosti na zadaném pravděpodobnostním rozdělení.

6.2 Poziční průvodce

Postup vytváření testovacího scénáře.

4. Nastavení globálních vlastností
 - Umístění textových stimulů – střed obrazovky.
 - Velikost písma – 30.
- Vložení textových stimulů
 - Target
 - Stimul1 – black (7000ms).
 - None-target
 - Stimul2 – purple (2000ms).
 - Stimul3 – yellow (5000ms).
 - Stimul4 – blue (5000ms).
- Vytvoření grafu (viz Obr 6. 5)










Obr 6. 5 Graf vytvořený Pozičním průvodcem.

6.2.1 Zhodnocení funkčnosti testovacího scénáře

Průvodce vytváří plně funkční spustitelný graf. Rozmístění jednotlivých symbolů určuje pořadí zobrazování stimulů na obrazovce, které je sekvenční. U tohoto typu průvodce není omezen počet stimulů jako u Pravděpodobnostního.

6.3 Použité symboly

Použité symboly vizuálního programovacího jazyka PreSti vztahující se k vytvořeným testovacím scénářům.¹ (viz Tabulka 6. 1).

	Spoj Connector	Spojnice pro přenos dat mezi jednotlivými symboly.
	Spoušť Trigger	Spojnice mezi jednotlivými symboly. Spoušť určuje, kterou akci symbolu vykonat při reakci na událost jiného symbolu. Posouvá dění scénáře dále.
	Vstupní bod Entry point	Určuje, kde bude běh scénáře začínat
	Opakovač Repeater	Po spuštění své akce vyvolává opakovaně svoji událost.
	Rozdělovač Splitter	Na signál z příchozí akce reaguje vyvoláním jedné z událostí na pravé straně. Jedná se o analogii k příkazu switch.
	Diskrétní generátor Discrete generator	Ze zadaného seznamu hodnot (čísel nebo řetězců) vybere náhodně jednu dle pravděpodobností přiřazeným těmto hodnotám.
	Textový stimul Text stimulus	Umožňuje vkládání textových stimulů a nastavování jejich vlastností.

Tabulka 6. 1 Použité symboly.

6.4 Otestování uživatelského rozhraní uživateli

Uživatelské rozhraní je téměř stejné pro oba dva průvodce. Otestováno bylo rozhraní Pravděpodobnostního průvodce, které je oproti druhému komplexnější.

- Pro testování byly vybrány tři subjekty, které neměly žádné zkušenosti s vytvářením testovacích scénářů v PreSti .
- Každému byly předány základní informace potřebné k pochopení vytváření testovacího scénáře v oblasti ERP.

¹ Informace převzaty ze zdroje [6]

6.4.1 Průběh testování

1. Nejprve byly subjekty seznámeny se zadáním (viz 5.2 a 6.1.1).
2. Při spuštění Náhodného průvodce subjektem byl aktivován měřič času.
3. V průběhu testování byly v případě potřeby subjektu zodpovězeny pouze otázky týkající se obecně ERP experimentů.
4. Po vytvoření testovacího scénáře byl změřen výsledný čas.

Subjekt	Čas
1	4 min
2	8 min
3	4 min

Tabulka 6. 2 Výsledky subjektů.

6.4.2 Výsledky testování

Každý subjekt splnil zadání a vytvořil funkční testovací scénář. V průměru vychází vytvoření scénáře na 5 minut. Rozdíl v rychlosti vytváření scénáře bude nejspíše závislý na dovednostech daného subjektu. (viz Tabulka 6. 2). Při testování byly objeveny drobné nedostatky, které byly následně odstraněny. Například chybělo označení jednotky pro pravděpodobnostní hodnotu výskytu stimulů.

7 Zhodnocení

Rozšíření funkcionality PreSti splňuje požadavky uvedené v zadání. S pomocí Pravděpodobnostního a Pozičního průvodce je možné vytvořit v přijatelném čase základní testovací scénáře bez nutnosti znalosti vizuálního programovacího jazyka. Průvodce vygeneruje spustitelný graf, který je možné později upravit a také uložit. Uživatelé mohou využít tento graf i jako učební pomůcku při poznávání vizuálního programovacího jazyka.

Funkcionalita Pravděpodobnostního průvodce je bohužel omezena počtem stimulů 4. Problém je v symbolu RandomGenerator, který nyní omezuje funkcionalitu a to tím, že nedovoluje přidávat neomezený počet stimulů. Tento problém je možné částečně obejít použitím Sekvenčního průvodce. Hodnoty, které se dříve náhodně generovaly, budou muset být zadány ručně jako posloupnost stimulů. Náhodné pořadí zobrazování stimulů bude muset člověk nahradit sekvenčním pořadím. Mimo jiné bylo otestováno, že RandomGenerator je schopný generovat odpovídající náhodné hodnoty v závislosti na jejich pravděpodobnostním rozložení.

- Shrnutí
 - Vytváření testovacích scénářů s nastavitelným pravděpodobnostním výskytem stimulů (max. 4 stimuly).
 - Vytváření scénářů se sekvenčním pořadím stimulů (neomezený počet stimulů).
 - Možnost úpravy grafu (testovacího scénáře) po jeho vygenerování průvodcem.
 - Možnost uložení scénáře.
 - Jednoduchost a intuitivnost uživatelského rozhraní.
 - Dostupná nápověda při vytváření scénářů.
 - Možné pozicování textu symbolu TextStimulus (9 poloh).
 - Oprava funkčnosti hlavního okna aplikace.

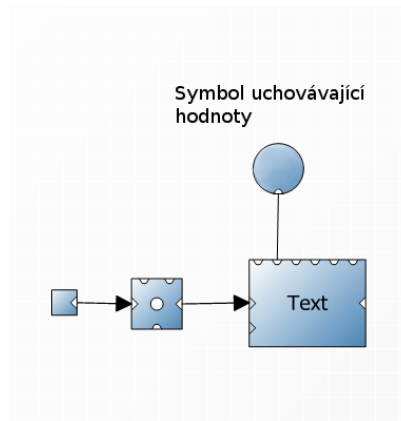
7.1 Návrhy na budoucí vylepšení

7.1.1 Interstimulus interval

V ERP experimentech je důležitou součástí testovacích scénářů. Je to časový úsek, který se nachází mezi koncem jednoho stimulu a začátkem druhého. V PreSti přímá funkcionalita intervalu není implementována. Existuje množina postupů, kterou lze interval nahradit, ale ta není schopna obsáhnout všechny případy jeho užití.

7.1.2 Symbol uchovávající hodnoty

Symbole reprezentující stimuly obsahují datové porty, díky kterým jsou schopny přijímat data. V případě použití stimulů v bloku působnosti symbolu Repeater by mohlo být efektivní postupně předávat předem připravené hodnoty symbolům a tím měnit jejich vlastnosti. Funkce by byla postavená na podobném principu fungování symbolů náhodných generátorů. Například by mohlo být možné každému cyklu Repeatru předávat TextStimulu hodnotu z předem definované množiny (viz Obr. 7. 1).



Obr. 7. 1 Příklad použití symbolu.

7.1.3 Úprava symbolu RandomGenerator

Pro vytváření obvyklých ERP experimentů je nutné, aby symbol byl schopen vygenerovat hodnotu a předat ji více symbolům. Nyní RandomGenerator generuje a zároveň předává hodnoty během procesu výměny dat, což znemožňuje předání jedné vygenerované hodnoty více symbolům. Tato skutečnost ovlivňuje větvení testovacího scénáře. Nyní omezuje funkčnost testovacího scénáře generovaného Pravděpodobnostním průvodcem.

7.1.4 SoundStimulus

Petr Háka ve své diplomové práci [7] přidal do množiny symbolů SoundStimulus. Symbol umožňuje přehrávat zvuky. V čase realizace své práce Petra Háka uvedl, že XNA nenabízel technologie schopné načítat zvuky určené pro přehrávání během spuštění aplikace. To znemožňuje PreSti vyrovnat se komerčním softwarům. Více informací [7].

7.1.5 Log soubor

Komerční softwary standardně poskytují zaznamenávání informací o průběhu testovacího scénáře do logovacího souboru. Tato funkčnost v PreSti chybí. Bylo by vhodné tuto funkcionalitu doplnit.

7.1.6 Chyba při navazování spojení symbolem Trigger

Během používání PreSti jsem objevil opakující se chybu. Chybu jsem zaregistroval při propojování symbolu EntryPoint s více TextStimuly. Je velice pravděpodobné, že chyba nastává obecně při propojování symbolů Triggerem. Chyba nastává bez závislosti na pořadí.

- 1) Propojení EntryPoint s 1. TextStimulem.
- 2) Propojení EntryPoint s 2. TextStimulem.
- 3) Propojení EntryPoint s 3. TextStimulem.
- 4) Chyba – zmizení spojení z bodu 2.

8 Závěr

Bakalářská práce se zabývá vylepšením softwaru PreSti, který je určen pro prezentaci stimulů. Aplikace používá k vytváření testovacích scénářů vizuální programovací jazyk, čímž se liší od ostatních softwarů. Cílem práce je PreSti vylepšit a tím ho přiblížit k úrovni komerčních softwarů.

V úvodu je čtenář seznámen se základními pojmy, které se vyskytují v oblasti ERP experimentů. Čtenář je informován o tom, co je ERP experiment, jakým způsobem se měří mozková aktivita a jak se dále zpracovávají získané signály z EEG zařízení. V této části je také vysvětleno, co jsou softwary pro prezentaci stimulů a k čemu jsou používány. Následuje další část, která se těmto softwarům věnuje podrobněji. Zde jsou čtenáři podrobně popsány funkce, vlastnosti, výhody a nevýhody jednotlivých komerčních softwarů. V závěru části je provedeno porovnání těchto softwarů s PreSti.

V další části dokumentu je zhotovena specifikace požadavků na vylepšení funkčnosti. Na základě této specifikace práce pokračuje návrhem přídatného modulu a jeho implementací. Implementovanou funkčnost reprezentuje průvodce pro vytváření testovacích scénářů a také další dodatečná vylepšení. PreSti pro vytváření testovacích scénářů používá vizuální programovací jazyk. Uživatel, který nezná tento jazyk, nemá možnost vytvořit základní testovací scénář. Z tohoto důvodu byl vytvořen průvodce, který umožňuje vytvářet základní scénáře bez nutnosti znalosti vizuálního programovacího jazyka PreSti. Průvodce je určen pro vytváření dvou typů testovacích scénářů pravděpodobnostního a pozičního. Pravděpodobnostní má nyní částečně omezenou funkcionalitu, která je způsobena nesprávnou funkčností jednoho symbolu vizuálního programovacího jazyka. Uživatel je schopný se v uživatelském rozhraní průvodce orientovat a s jeho pomocí vytvořit testovací scénář v řádech minut. Po vytvoření je možné dále upravovat scénář a také ho uložit.

Z předchozí kapitoly zhodnocení (viz 7) vyplývá, že aplikace byla úspěšně rozšířena o další funkcionalitu. PreSti bude potřebovat ještě další vylepšení. V kapitole 7.1 je sepsán seznam možného budoucího vylepšení aplikace.

Na závěr je nutné podotknout, že práce splňuje všechny požadavky uvedené v zadání. Uživatel je schopen bez znalosti vizuálního programovacího jazyka vytvořit testovací scénář.

Literatura

- [1] NIEDERMEYER, Ernst a LOPES DA SILVA, Fernando. *Electroencephalography: Basic Principles, Clinical Applications, and Related Fields*. Lippincot : Williams & Wilkins, 2004.
- [2] Alpha waves during Natural Stress Relief meditation. *ElettroEncefaloGrafia Quantitativa*. [Online] [Citace: 13. 7. 2010.]. Dostupné z: <http://www.eeg.it/eeg/nsr.php>.
- [3] GOLIN, E. J. a REISS, S. P. *The Specification of Visual Language Syntax*. Journal of Visual Languages and Computing. 1990, Sv. 1, 2.
- [4] LUCK, Stephen J. *An Introduction to the Event-Related Potential Technique*. California : The MIT Press, 2005. ISBN:978-0-262-62196-0.
- [5] SLAVÍČKOVÁ, A. BRUNOVSKÝ, M a MOHR, P. *Kognitivní evokované potenciály v klinické praxi a experimentu*. Psychiatrie. 4(1), 2010.
- [6] JAROŠ, Petr. *Software pro prezentaci stimulů*. Plzeň, 2010. Diplomová práce. Západočeská univerzita v Plzni. Fakulta aplikovaných věd.
- [7] HÁKA, Petr. *Software for presentation of stimuli*, Plzeň, 2011. Diploma Thesis. University of West Bohemia. Faculty of Applied Sciences.
- [8] DirectRT. *Empirisoft Corporation, NYC*. [Online] [Citace: 5. 5. 2013]. Dostupné z: <http://www.empirisoft.com/directrt.aspx>.
- [9] Presentation. *NeuroBehavioral Systems*. [Online] [Citace 5. 5. 2013]. Dostupné z: http://www.neurobs.com/menu_presentation/menu_features/features_overview.
- [10] Paradigm. *Perception Research Systems Inc*. [Online] [Citace 5. 5. 2013]. Dostupné z: <http://www.paradigmexperiments.com/features.html>.

Přehled zkratk

CSV – jednoduchý souborový formát, který je určen pro výměnu tabulkových dat. Data jsou standardně oddělena čárkami.

EEG - (neboli elektroencefalogram) je záznam časové změny elektrického potenciálu způsobeného mozkovou aktivitou.

ERP - Evokovaný potenciál, Event Related Potential) je reakce mozku na vnější nebo vnitřní stimul.

PreSti – software pro prezentaci stimulů v ERP experimentech. Vytvořený na Západočeské univerzitě v roce 2010 Petrem Jarošem.

TTL – (Transistor Transistor logic) je standardem používaným pro implementaci digitálních (také logických) integrovaných obvodů.

UML – (Unified Modeling Language) je grafický jazyk pro vizualizaci, specifikaci, navrhování a dokumentaci programových systémů.

XNA – sada vývojářských nástrojů určená především pro vývoj herních aplikací.

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Uživatelská příručka programu
PreSti

Obsah

1	Úvod.....	1
2	Instalace.....	2
2.1	Systémové požadavky	2
2.2	Příprava počítače	2
2.3	Instalace ovladače paralelního portu	2
2.4	Spuštění programu	4
2.5	Kompilace ze zdrojového kódu	4
3	Funkce programu.....	5
3.1	Rozložení ovládacích prvků	5
3.2	Práce s testovacími scénáři	5
3.2.1	Otevření testovacího scénáře.....	6
3.2.2	Spuštění testovacího scénáře	6
3.2.3	Spuštění testovacího scénáře v testovacím režimu	6
3.2.4	Vytvořit podgraf.....	6
3.2.5	Odstranit podgraf.....	6
3.2.6	Uložení testovacího scénáře	6
3.2.7	Otevření prázdného testovacího scénáře	6
3.2.8	Zavření testovacího scénáře	6
3.2.9	Ukončení programu.....	6
3.3	Vytváření a editace testovacích scénářů	6
3.3.1	Vytváření scénářů průvodcem.....	8
3.3.2	Symboly k vytváření scénářů	10
3.4	Rozšiřování funkčnosti programu	13
3.4.1	Vytvoření nového modulu.....	13

Tento dokument vznikl úpravou následujících uživatelských manuálů, jejich spojením a rozšířením:

1. Uživatelská příručka softwaru PreSti ⁱ.
2. Uživatelská příručka rozšíření softwaru PreSti v anglickém jazyce (přeložena do češtiny) ⁱⁱ.
3. Rozšíření dokumentu o kapitulu Vytváření scénářů průvodcem (viz 3.3.1).

1 Úvod

PreSti je program pro vytváření testovacích scénářů, které se používají v neuroinformatických ERP experimentech. V rámci testovacích scénářů umožňuje prezentovat testovanému subjektu stimuly a komunikovat s přístrojem na měření EEG signálů z důvodu synchronizace počátků stimulů a následné odezvy mozku.

2 Instalace

2.1 Systémové požadavky

- Operační systém Microsoft Windows XP SP2 a novější.
- Knihovny .NET Framework 4.0 nebo novější.
- Microsoft XNA Framework 4.0 nebo novější.
- Volitelně paralelní port.

2.2 Příprava počítače

Program potřebuje nutně ke svému běhu knihovnu .NET Framework 4 a XNA Framework 4.

1. .NET framework 4 - <http://www.microsoft.com/en-us/download/details.aspx?id=20914>
2. XNA framework 4 - <http://www.microsoft.com/en-us/download/details.aspx?id=23714>

Instalace je rychlá a intuitivní. V průběhu si z internetu stáhne potřebné soubory. Není potřeba restartovat systém.

2.3 Instalace ovladače paralelního portu

Ovladač paralelního portu je potřeba nainstalovat v případě potřeby komunikace s EEG přístrojem. V opačném případě je možné spustit program i bez této instalace. Pro správnou funkčnost ovladače je třeba, aby byl v počítači přítomen paralelní port.

Následuje postup instalace paralelního portu na 32 bitový operační systém. Stejným způsobem by bylo možné je nainstalovat i na 64 bitový systém, pokud by byl řádně podepsán certifikátem vydaným uznávanou certifikační autoritou. Toto je způsobeno zvýšeným zabezpečením 64 bitových operačních systémů firmy Microsoft. Tématem podepisování ovladačů se zabývá kapitola 2.5.5 Podepisování ovladačů v diplomové práci.

Postup pro instalaci ovladače (postup pro Windows 7):

1. Otevřít Ovládací panely.
2. Vybrat položky Hardware a zvuk - > Správce zařízení.
3. V tomto kroku může systém požadovat administrátorská oprávnění.
4. Kliknout pravým tlačítkem na kořen stromu na obrazovce. Je označen názvem počítače.
5. Vyberte položku Přidat starší hardware.
6. Následující obrazovku potvrďte tlačítkem Další.
7. Vyberte položku Nainstalovat Hardware, který vyberu ze seznamu.
8. Tuto obrazovku potvrďte tlačítkem Další.

9. Pokračujte dále tlačítkem Z disku.
10. Tlačítkem Procházet vyberte soubor parallelport.inf. Ve stejné složce musí být obsažen i soubor parallelport.sys a WdfCoInstaller01009.dll.
11. Tlačítka Otevřít, poté OK, Další a Další spustíte instalaci.
12. Na dotaz o zabezpečení odpovězte Přesto nainstalovat tento software ovladače.
13. Po nějaké chvíli se dokončí instalace. Poté ji ukončete tlačítkem Dokončit.

Postup odinstalace ovladače (postup pro Windows 7):

1. Kroky 1 – 4 jsou shodné jako při instalaci.
2. V stromu zařízení vyhledejte zařízení Systémová zařízení -> Parallel Port Driver (KMDF).
3. Pravým tlačítkem na tomto zařízení vyberte volbu Odinstalovat.
4. Potvrďte tlačítkem OK.

Postup instalace ovladače (postup pro Windows XP)

1. Otevřít Ovládací panely.
2. Vybrat položku Tiskárny a jiný hardware a nalevo v seznamu položku Přidat hardware.
3. V otevřeném okně pokračujte tlačítkem Další.
4. Vyberte položku Ano, hardware byl již připojen a pokračujte tlačítkem Další.
5. Ze seznamu vyberte poslední položku Přidat nové hardwarové zařízení a pokračujte tlačítkem Další.
6. Vyberte položku Nainstalovat hardware, který vyberu ze seznamu a pokračujte tlačítkem Další.
7. Pokračujte od bodu 8 z postupu na instalaci ve Windows 7.

Postup odinstalace ovladače (postup pro Windows XP):

1. Otevřít Ovládací panely.
2. Vybrat položku Tiskárny a jiný hardware a nalevo v seznamu položku Systém.
3. Vyberte záložku Hardware.
4. Otevřete Správce zařízení stejnojmenným tlačítkem.

Alternativní postup pro instalaci:

1. Spustit program install.bat kliknutím pravým tlačítkem a výběrem volby Spustit jako administrátor.

Alternativní postup pro odinstalaci:

1. Spustit program uninstall.bat kliknutím pravým tlačítkem a výběrem volby Spustit jako administrátor.

2.4 Spuštění programu

Program se spouští ze souboru PreStiApplication.exe.

Pokud bude program připojen k přístroji na měření EEG signálu a bude tedy komunikovat po paralelním portu, je potřeba, aby byl spuštěn s oprávněními administrátora:

- Pravým tlačítkem kliknout na PreStiApplication.exe.
- Vybrat položku Spustit jako Administrátor.

2.5 Kompilace ze zdrojového kódu

Ke kompilaci ze zdrojových kódů je potřeba nejprve instalace vývojového prostředí a nástrojů:

- Visual Studio 2010 a novější.
- NShape (<http://code.google.com/p/nshape/downloads/list>).
- Nejnovější Windows Driver Kit.
- Nejnovější DirectX SDK.
- Nejnovější Windows SDK.

Poslední tři položky jsou zdarma ke stažení na stránkách firmy Microsoft. Alternativně je možné použít instalační soubory na přiloženém DVD.

Poté je potřeba zkopírovat zdrojový kód z CD, popřípadě z SVN na adrese:

<https://presti.svn.codeplex.com/svn>

Ke stažení z SVN je možné použít například program TortoiseSVN¹. Po otevření souboru PreSti.sln ve Visual Studiu, je možné program ihned zkompileovat klávesou F6. Je možné vybrat na mezi sestavením debug obsahující informace pro ladění programu nebo release určeným k distribuci programu. Program je závislý na knihovnách, které jsou dostupné pouze ve verzích pro 32 bitový systém. Architektura by měla být vždy nastavena na X86, nehledě na cílovou platformu (32 nebo 64 bitový systém).

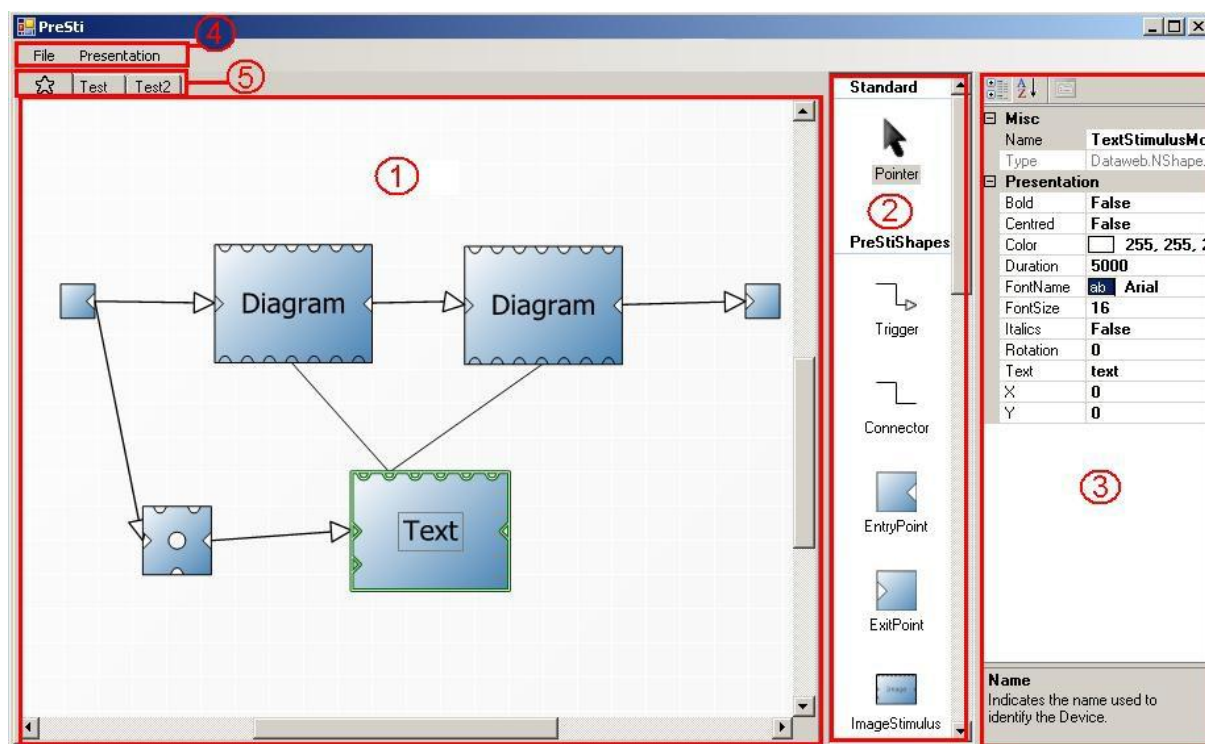
¹ SVN klient pro systém microsoft Windows. <http://tortoisesvn.tigris.org/>.

3 Funkce programu

3.1 Rozložení ovládacích prvků

Rozložení ovládacích prvků v hlavním okně programu je znázorněno na obrázku 3.1. Okno se skládá ze čtyř částí:

1. Okno pro vytváření a editaci testovacích scénářů pomocí diagramů ve vizuálním programovacím jazyce.
2. Panel dostupných symbolů k použití v diagramu.
3. Panel vlastností symbolů umístěných v diagramu.
4. Menu pro ovládání programu.
5. Panel diagramů



Obr 3. 5 Hlavní okno programu PreSti

3.2 Práce s testovacími scénáři

Testovací scénáře je možné vytvářet a upravovat symboly vizuálního programovacího jazyka, které jsou dostupné v panelu symbolů (viz panel 2 na Obr 3. 5), jenž je umístěný v pravé části okna. K vložení symbolu do diagramu přetáhněte symbol technologií drag and drop do okna diagramu. Každý z uvedených symbolů má svoji funkčnost, kterou reprezentují nastavitelné atributy (viz panel 3 na Obr 3. 5). Seznam symbolů je uvedený v kapitole 3.3.2 . Přepínací panel (panel 5 na Obr 3. 5) je určen pro manipulaci s podgrafy. Položka v přepínacím panelu reprezentuje okno podgrafu. Funkcionalita podgrafu je definována v tomto okně. Úkony s testovacími scénáři je možno provádět v hlavním menu programu.

3.2.1 Otevření testovacího scénáře

Otevření testovacího scénáře se provádí v menu položkou File -> Open. Testovací scénář je vždy uložen v XML souboru. Na Příloženém DVD je sada ukázkových testovacích scénářů.

3.2.2 Spuštění testovacího scénáře

Hotový testovací scénář se spouští položkou v menu Presentation -> Run nebo klávesou F5. Všechny objekty testovacího scénáře jsou nejprve načteny do paměti a poté je teprve program spuštěn. Přednačtení objektů předem zaručuje přesné zobrazení stimulů a plynulý běh scénáře. Spuštění testovacího scénáře je potřeba potvrdit klávesou mezerník.

3.2.3 Spuštění testovacího scénáře v testovacím režimu

Vyvíjený testovací scénář je možné si spustit místo režimu na celou obrazovku pouze v okně. Okno je doplněno o časový údaj od spuštění scénáře a tak je možné sledovat, zda scénář probíhá podle představ. Spuštění scénáře proběhne až po stisku klávesy mezerník.

3.2.4 Vytvořit podgraf

Pro přidání nového podgrafu, vyberte File -> Create Subdiagram. Tato posloupnost operací otevře dialog pro pojmenování podgrafu.

3.2.5 Odstranit podgraf

Pro odstranění podgrafu, vyberte File -> RemoveSubdiagram. Hlavní diagram označený hvězdičkou není možné odstranit.

3.2.6 Uložení testovacího scénáře

Testovací scénář je možné uložit do XML souboru položkou File -> Save nebo File -> Save As... v menu programu. Alternativně je pro tyto příkazy možno použít klávesové zkratky Ctrl+S a Ctrl+Shift+S.

3.2.7 Otevření prázdného testovacího scénáře

Otevření nového testovacího scénáře se provádí v menu položkou File -> New. Před otevřením nového scénáře bude zobrazena výzva k uložení stávající práce.

3.2.8 Zavření testovacího scénáře

Testovací scénář je možno po ukončení práce uzavřít položkou File -> Close v menu programu.

3.2.9 Ukončení programu

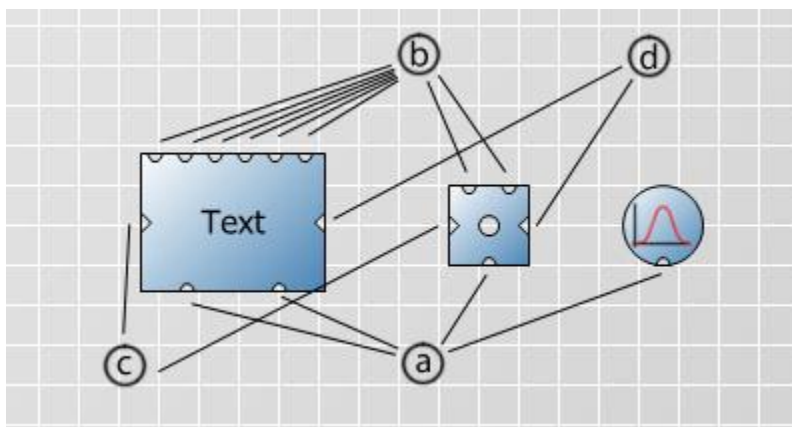
Ukončit celý běh programu je možné položkou File -> Exit v menu, uzavřením celého okna nebo klávesovou zkratkou ALT+F4.

3.3 Vytváření a editace testovacích scénářů

Vytváření a editace testovacích scénářů se provádí návrhem diagramu. V diagramu mohou být použity pouze symboly z panelu dostupných symbolů. Symboly se do okna pro editaci umisťují vybráním konkrétního symbolu v panelu dostupných symbolů a poté označením

místa pro vložení v okně pro editaci levým tlačítkem myši. Každý symbol vyjadřuje nějakou ucelenou činnost a má různé množství vlastností vztahujících se k této činnosti. Význam jednotlivých symbolů a jejich vlastností je popsán v kapitole kapitole 3.3.2. Vlastnosti symbolu je možné nastavit v panelu vlastností. Všechny symboly mimo Triggeru a Connectoru, které jsou určeny ke spojování symbolů, jsou po stranách osazeny různým počtem portů. Existují čtyři druhy portů:

- a) Výstup hodnoty z objektu (Getter)
 - Zpřístupňuje hodnoty objektu navenek, aby tyto hodnoty mohly využít ostatní objekty. Má značku půlkruhu a bývá osazen na spodní části symbolu.
- b) Vstup hodnoty do objektu (Setter)
 - Vlastnosti u symbolů mohou být nastaveny staticky před spuštěním testovacího scénáře nebo se mohou měnit až při běhu scénáře na základě stavu okolních objektů. K dynamickému nastavení vlastnosti se používá právě setter. Má značku půlkruhu a bývá osazen na svrchní části symbolu.
- c) Událost (Event)
 - Používá se k oznámení určité události okolí. Pomocí událostí je možné interagovat s ostatními symboly. Má značku trojúhelníku a bývá umístěna na pravé části symbolu.
- d) Akce (Action)
 - Umožňuje reagovat na událost vzniklou v okolí a spustit nějakou vlastní činnost. Má značku trojúhelníku a bývá umístěna na levé části symbolu.



Obr 3. 6 Osazení porty u jednotlivých symbolů

Ukázky symbolů jsou uvedeny na obrázku 3.2, písmena uvedená v obrázku odpovídají seznamu portů výše. U portů stejného typu na obrázku není možné rozlišit, jakou funkci provádí. Tento problém je v programu vyřešen kontextovou nápovědou, kdy po přejetí portu myší se zobrazí jeho význam.

Jednotlivé symboly se spojují právě na místě výše uvedených portů speciálními symboly:

- Spojení (Connection)

- Používá se k propojení getterů a setterů a umožňuje tím přenášet hodnoty mezi objekty. Přenos hodnoty iniciuje vždy objekt s setterem na své straně spojení. V diagramu je spojení označeno plnou čarou.
- Spoušť (Trigger)
 - Slouží k propojení akcí a událostí. Spoušť určuje akci, kterou bude reagováno na událost. Nejběžnějším použitím spouště je reakce na událost ukončení stimulu spuštěním akce start jiného stimulu. Spoušť má jedinou vlastnost – zpoždění, kterým lze určit prodlevu mezi vyvoláním události a vykonáním akce. V diagramu je spoušť označena šipkou ve směru od události k obslužné akci.

V případě, že bylo do diagramu umístěno spojení nebo spoušť a nebylo řádně propojeno mezi správnými typy portů, bude automaticky odstraněno.

Testovací scénáře se vytváří pouhým spojováním symbolů použitím spojení a spouští a nastavením vlastností symbolům.

3.3.1 Vytváření scénářů průvodcem

Postup:

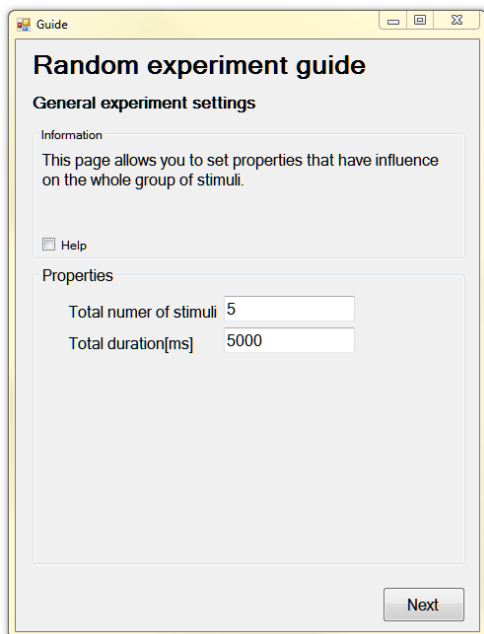
1. Spustíte průvodce - File -> New -> With guide.
2. Vyberte jednu z možností:
 - Random
 - Průvodce umožňuje vytvořit jednoduchý experiment, ve kterém lze nastavit pravděpodobnost stimulů poskytovanou náhodným generátorem.
 - Sequential
 - Průvodce poskytuje nástroje pro vytvoření sekvenčního experimentu. Pozice v seznamu stimulů definuje jejich pořadí.

Pravděpodobnostní experiment

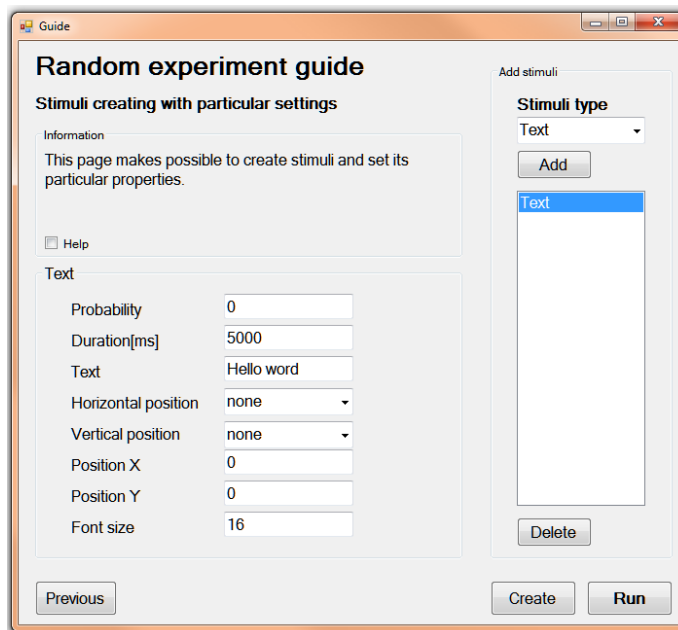
Strana 1

Náhodný průvodce pracuje se skupinou stimulů. Stránka poskytuje nastavení vlastností této skupiny. (viz Obr 3. 7)

- Total number stimuli – počet stimulů, které se budou nacházet ve skupině.
- Total duration – prezentace jednoho stimulu spolu s interstimulus intervalem (doba mezi koncem stimulu a začátkem dalšího stimulu). Total duration = stimulus interval + interstimulus interval.



Obr 3. 7 Náhodný průvodce - strana 1



Obr 3. 8 Náhodný průvodce - strana 2

Postup:

1. Nastavte Total number stimuli.
2. Nastavte Total duration.
3. Stiskněte tlačítko Next.

Strana 2

Poskytuje nástroje k vytvoření stimulů, nastavení vlastností jako je pravděpodobnost, doba trvání atd. Pro vytvoření stimulu vyberte jeden ze 3 typů – Text, Obrázek (Image), Zvuk (Sound). Po přidání stimulu do seznamu můžete nastavit nebo změnit jeho vlastnosti. (viz Obr 3. 8)

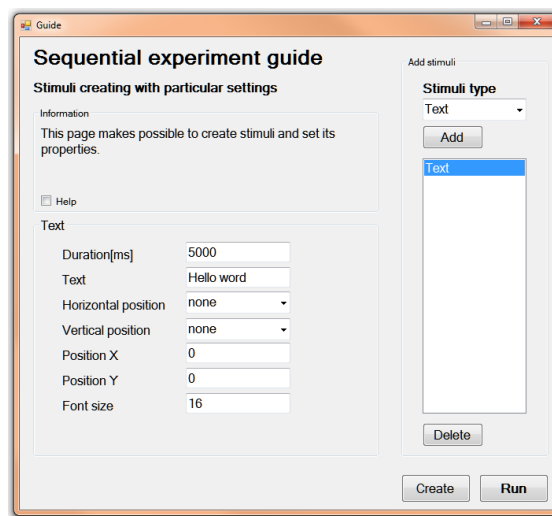
Postup

1. Vyberte jeden typ stimulu v rozevřacím seznamu, který je umístěn v panelu Add stimuli (pravá strana okna).
2. Potvrďte svůj výběr stisknutím tlačítka Add.
3. Nastavte vlastnosti již přidáných stimulů: označte stimul v seznamu, který je umístěn pod tlačítkem Add.
4. Vlastnosti jednotlivých stimulů jsou zobrazeny v pravé dolní části panelu Properties. Jestliže chcete změnit nebo nastavit vlastnost stimulu, klikněte na textové pole a zadejte hodnotu.
5. Spusťte experiment a vytvořte graf stisknutím tlačítka Run. V případě, že chcete vytvořit jen graf, stiskněte Create.

Poziční experiment

Strana 1

Poskytuje nástroje k vytvoření stimulů, nastavení vlastností jako je pozice, doba trvání atd. Pro vytvoření stimulu vyberte jeden ze 3 typů – Text, Obrázek (Image), Zvuk (Sound). Po přidání stimulu do seznamu můžete nastavit nebo změnit jeho vlastnosti. Nakonec můžete spustit experiment. Pořadí běhu stimulů je stejné, jako je pořadí stimulů v seznamu. (viz Obr 3. 9)














Obr 3. 9 Sekvenční průvodce - strana 1









- Postup


1. Vyberte jeden typ stimulu v rozevřacím seznamu, který je umístěn v panelu Add stimuli (pravá strana okna).
2. Potvrďte svůj výběr stisknutím tlačítka Add.
3. Nastavte vlastnosti již přidávaných stimulů: označte stimul v seznamu, který je umístěn pod tlačítkem Add.
4. Vlastnosti jednotlivých stimulů jsou zobrazeny v pravé dolní části panelu Properties. Jestliže chcete změnit nebo nastavit vlastnost stimulu, klikněte na textové pole a zadejte hodnotu.
5. Spusťte experiment a vytvořte graf stisknutím tlačítka Run. V případě, že chcete vytvořit jen graf, stiskněte Create.

3.3.2 Symboly k vytváření scénářů

V tabulce 3.1 jsou uvedeny symboly, které jsou standardní součástí programu PreSti. Tabulka obsahuje pouze popis funkce symbolu a výpis zajímavých vlastností. Detailní popis jednotlivých vlastností je popsán v kontextové nápovědě v panelu vlastností.

	Spoj Connection	Spojnice pro přenos dat mezi jednotlivými symboly. Spojuje porty setter a getter.
	Spoušť Trigger	Spojnice mezi jednotlivými symboly. Spoušť určuje, kterou akci symbolu vykonat při reakci na událost jiného symbolu. Posouvá dění scénáře dále.
	Vstupní bod Entry Point	Určuje, kde bude běh scénáře začínat. Vstupní bod musí obsahovat každý testovací scénář
	Textový stimul Text Stimulus	Textový stimul umožňuje zobrazit na obrazovce text. Stimul má akce spuštění, předčasného ukončení a událost při ukončení. Má různá množství vlastností pro nastavení fontu, pozice a natočení textu. Vlastnosti jsou přístupné přes gettery a settery.
	Zvukový stimul Sound Stimulus	Zvukový stimul dovoluje začlenit zvuk do testovacích scénářů. Jeho parametry jsou – doba trvání a jméno zvuku, které se nachází v seznamu dostupných zvuků.
	Obrazový stimul Image Stimulus	Obrazový stimul umožňuje zobrazit obrázek ze souboru na obrazovce. Stimul má akce spuštění, předčasného ukončení a událost při ukončení. Má různá množství vlastností nastavení velikosti, pozice, natočení obrázku a další. Vlastnosti jsou přístupné přes gettery a settery.
	Podgraf Subdiagram	Symbol má pouze jeden parametr – jméno. Parametr propojuje symbol s oknem, které obsahuje funkčnost podgrafu. Všechny gettery a settery symbolu reprezentují vstupní a výstupní hodnoty definované v podgrafu.
	Setter	Symbol použitelný pouze v podgrafech. Je určen k předávání vstupní hodnoty z portu symbolu Podgraf do jeho grafu. Parametr index definuje, který port symbolu Podgraf je se setterem svázán.
	Getter	Symbol použitelný pouze v podgrafech. Je určen k předávání výstupní hodnoty z grafu symbolu Podgraf na port symbolu. Parametr index definuje, který port symbolu Podgraf je s getterem svázán.
	Exit Point	Symbol použitelný pouze v podgrafech. Definuje konec podgrafu.
	Generátor normálního rozd. Normal Distr. Generator	Generátor normálního rozdělení náhodných čísel má pouze jediný getter, ve kterém vrací vygenerovanou hodnotu. V jeho vlastnostech je možné nastavit střední hodnotu, rozptyl a další parametry rozdělení.

	Generátor exp. Rozdělení Exponential Distr. Generator	Generátor exponenciálního rozdělení náhodných čísel má pouze jediný getter, ve kterém vrací vygenerovanou hodnotu. V jeho vlastnostech je možné nastavit střední hodnotu, koeficient strmosti a další parametry rozdělení.
	Generátor rovn. Rozdělení Uniform Distr. Generator	Generátor rovnoměrného rozdělení náhodných rozdělení má pouze jediný getter, ve kterém vrací vygenerovanou hodnotu. V jeho vlastnostech je možné nastavit počátek a konec intervalu pro generování čísel.
	Diskrétní gen. Discrete Gen.	Generátor diskrétního rozdělení náhodných rozdělení má pouze jediný getter, ve kterém vrací vygenerovanou hodnotu. Ze zadaného seznamu hodnot (čísel nebo řetězců) vybere náhodně jednu - dle pravděpodobností přiřazeným těmto hodnotám.
	Výstup Output	Výstup na paralelní port. Umožňuje určit hodnotu výstupu.
	Vstup z myši Mouse Input	Umožňuje přidat do testovacích scénářů reakci uživatele zadanou pomocí myši. Vyvolává událost v případě provedené akce a z getteru je možné určit pozici myši. Ve vlastnostech je možné nastavit, na která tlačítka bude reagováno.
	Vstup z klávesnice Keyboard Input	Umožňuje přidat do testovacích scénářů reakci uživatele zadanou z klávesnice. Vyvolává událost v případě provedené akce a z getteru je možné určit hodnotu stisknuté klávesy v podobě znaku nebo čísla.
	Rozdělovač Splitter	Na signál z příchozí akce reaguje vyvoláním jedné z událostí na pravé straně. Výběr události je dán hodnotou ze setteru. Příchozí hodnotu porovná s tabulkou hodnot přiřazených jednotlivým událostem. Jedná se o analogii k větvení programu konstrukcí switch. Poslední z událostí je provedena v případě, že žádná jiná událost nebyla vybrána (default větev konstrukce switch).
	Směšovač Mixer	V případě spuštění jakékoliv z akcí je vyvolána jediná událost směšovače. V getteru je poté k dispozici hodnota dle tabulky přiřazující akcím na levé straně libovolné hodnoty. Tabulku je možné nastavit ve vlastnostech.

	<p>Opakovač</p> <p>Repeater</p>	<p>Po spuštění své akce vyvolává opakovaně svojí událost. Počet událostí a doba opakování je dána vlastnostmi ze setterů. Z getteru je možné načíst číslo označující počet provedených cyklů.</p>
---	---	---

Tabulka 3. 2 Seznam symbolů

3.4 Rozšiřování funkčnosti programu

Možnosti při vytváření diagramů je možno rozšířit přidáním dalších symbolů. Modul pro rozšíření je DLL soubor obsahující knihovnu symbolů. Jsou dvě možnosti jak získat nový modul: Využít stávajících modulů volně ke stažení naprogramované někým jiným, nebo naprogramovat modul vlastní.

Modul se v programu načítá ze složky modules ve stejném adresáři jako je umístěn program PreSti. Pokud tato složka neexistuje, je potřeba ji vytvořit. Pro instalaci modulu je potřeba jej umístit do složky modules a otevřít nový testovací scénář. Není nutné program restartovat. V případě, že je otevírán testovací scénář, který využívá symbolů z nějakého modulu, je nejprve prohledána složka modules. Pokud patřičný modul není nalezen, je uživatel upozorněn na chybějící modul. Dokud nepřiloží správný modul k programu, nemá možnost testovací scénář otevřít.

3.4.1 Vytvoření nového modulu

Modul je standardní DLL soubor obsahující třídy definující symbol, jeho vlastnosti a funkčnost. Třídy musí implementovat daná rozhraní, jinak nebude symbol načten programem PreSti.

Ve složce se zdrojovým kódem je složka PreStiModuleSample obsahující projekt se vzorovým modulem.

Každý symbol je definován sadou třech tříd (třídy mohou být libovolného jména, zde jsou uvedeny třídy z projektu PreStiModuleSample):

- SampleStimulusSymbol
 - Třída definuje vzhled symbolu. Třída musí implementovat rozhraní Shape frameworku NShape, aby se zobrazila v panelu symbolů. Framework obsahuje velké množství abstraktních tříd, které již částečně rozhraní Shape implementují. Příkladem abstraktních tříd jsou RectangleBase nebo CircleBase.
 - Druhé rozhraní, které musí třída implementovat je rozhraní ISymbol. Implementace rozhraní umožní převod symbolu do vnitřní reprezentace v PreSti a kontextovou nápovědu.
- SampleStimulusModule
 - Defínuje vlastnosti symbolu tak, aby je bylo možné editovat v panelu vlastností a uložit spolu s diagramem do souboru. Musí implementovat rozhraní ModelObjectBase frameworku NShape.
- SampleStimulusComponent

- Vnitřní reprezentace symbolu používaná při spouštění testovacího scénáře. V této třídě je definována funkčnost symbolu. Musí být potomkem abstraktní třídy `BaseComponent`, pokud bude systém používat pouze porty `setter` a `getter` (například generátory náhodných zobrazení). Spustitelné symboly (jako jsou stimuly), které potřebují využívat portů akce a událost musí být potomkem `RunnableComponent`, která je následníkem třídy `BaseComponent`.

Každý modul může obsahovat libovolné množství takto vytvořených symbolů. Každý ze symbolů je potřeba zaregistrovat, aby o něm program věděl. Registrace symbolu se provádí ve speciální třídě `NShapeLibraryInitializer` v modulu společném pro všechny symboly. Konkrétní obsah třídy je uveden v projektu `PreStiSampleModel`.

ⁱ JAROŠ, Petr. *Software pro prezentaci stimulů*. Plzeň, 2010. Diplomová práce. Západočeská univerzita v Plzni. Fakulta aplikovaných věd.

ⁱⁱ HÁKA, Petr. *Software for presentation of stimuli*, Plzeň, 2011. Diploma Thesis. University of West Bohemia. Faculty of Applied Sciences.