Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií Ilkovičova 2, 842 16 Bratislava 4

Trojdimenzionálne UML

Dokumentácia k inžinierskemu dielu

Tím č. 2 Členovia tímu: Bc. Brndiarová Gabriela, Bc. Štajer Andrej, Bc. Valko Andrej, Študijný program: IS/SI Ročník: 1. Ing. Predmet: Tímový projekt Vedúci tímu: Ing. Polášek Ivan, PhD. Ak. rok: 2013/14

Bc. Kysel' Peter,Bc. Martoš Ivan,Bc. Štetiar Matej,Bc. Šuta Erik,Bc. Kolačkovský Tomáš (do 6. týždňa ZS)

Obsah

0	ObsahII			
1	Ú	Ĵvod	1	
	1.1	Celkový pohľad na projekt	1	
	1.2	Ciele projektu	1	
	1.3	Implementácia	2	
2	0	Ciele projektu – zimný semester	3	
3	Š	print č. 1	4	
3.1 Vytvorenie prezentačnej stránky		Vytvorenie prezentačnej stránky	4	
	3.2	Analýza problémovej oblasti	5	
	3.3	Spustenie existujúceho prototypu	5	
	3.4	Príprava aplikácie	5	
	3.5	Pridať čiaru života	6	
4	Š	Sprint č. 2	7	
	4.1	Pridať novú vrstvu	7	
	4.2	Zadať názov inštancie a názov triedy pre čiaru života	7	
	4.3	Zjednodušenie výberu čiary života	9	
	4.4	Pridať interakciu medzi objektmi	9	
	4.5	Zarovnanie čiar života	10	
	4.6	Uložiť a načítať diagram	10	
	4.7	Oprava chýb	11	
	Т	Fieň v obdĺžniku čiary života	11	
	P	Prerušovaná čiara presahuje vrstvu	12	
5	Š	Sprint č. 3	13	
	5.1	Vymazávanie objektov	13	
	5.2	Maximálna šírka čiary života	13	
	5.3	Pridanie atribútov pre prácu so šírkou a výškou elementov	14	
	5.4	Označenie čiar života ťahom myši (pre vloženie fragmentu)	14	
	5.5	Návrh fragmentu	15	
	5.6	Zalamovanie textu v čiarach života	15	
	5.7	Zarovnávanie čiar života	17	
	5.8	Ukladanie a načítavanie diagramov	19	
	5.9	Oprava chýb	19	
Chybné získavanie údajov z formulára na vytvorenie čiary života		Chybné získavanie údajov z formulára na vytvorenie čiary života	19	
Oprava chyby výberu elementu na skrytej vrstve		Oprava chyby výberu elementu na skrytej vrstve	19	
Nemožnosť vytvoriť čiaru života z kódu		Vemožnosť vytvoriť čiaru života z kódu	20	
6	Š	Sprint č. 4	21	
	6.1	Vytvorenie programovej infraštruktúry pre mazanie	21	
	S	Spustenie mazania	21	
	V	/ymazanie elementu	21	
	N	Mazanie asynchrónnych správ	21	
	6.2	Načítavanie diagramu z XML	22	
	6.3	Označenie čiar života ťahom myši – upravená logika	22	
	6.4	Oprava chýb	22	
	N	Vepredvídateľné miznutie fragmentu	22	
	U	Jrčenie pozície fragmentu na zadných vrstvách	22	
	Γ	Doplnenie referencií na interakcie do čiar života	23	

7 Šj	orint č. 5	. 24
7.1	Zvýraznenie obalu pri prechode cez čiaru života	. 24
7.2	Vymazanie aktuálneho modelu	. 24
7.3	Optimalizácia exportu a importu scény do/z XML	. 25
7.4	Vymazanie čiary života z diagramu	. 25
7.5	Oprava chýb	. 25
Fr	agment sa vykresľuje ako vodorovná čiara	. 25
Či	ara života sa zvýrazní aj keď sa nemá	. 26
V	kresľovanie fragmentu na prekrývajúcu sa vrstvu	. 26
Pı	i ukladaní scény program zamrzne	. 26
8 C	elkový pohľad na produkt po ZS	. 27
8.1	Prototyp ako celok	. 27
8.2	Architektúra produktu	. 28
8.3	Dátový model	. 30
La	yer (Vrstva)	. 31
Li	feline (Čiara života)	. 31
A	synchronous Message (Asynchrónna správa)	. 31
Fr	agment	. 32
9 Šj	print č. 6	. 33
9.1	Dokončovacie práce na funkcionalite vymazávania	. 33
9.2	Zarovnanie správ pri vkladaní	. 33
9.3	Zarovnanie čiar života	. 34
9.4	Oprava chýb	. 36
Μ	azanie vrstvy ako elementu	. 36
0	prava umiestnenia LayerBookmark po vymazaní scény	. 37
0	prava rozmerov tlačidiel – zamedzenie pretekania textu tlačidiel	. 37
0 10 Šj	prava rozmerov tlačidiel – zamedzenie pretekania textu tlačidiel print č. 7	37 . . 38
0 10 Šj 10.1	prava rozmerov tlačidiel – zamedzenie pretekania textu tlačidiel print č. 7 Preberanie fragmentu (Šprint 7,8,9)	37 38 38
0 10 Šj 10.1 10.2	prava rozmerov tlačidiel – zamedzenie pretekania textu tlačidiel print č. 7 Preberanie fragmentu (Šprint 7,8,9) Posúvanie prvkov diagramu pri vkladaní nových správ	37 38 38 39
10 Šj 10.1 10.2 10.3	prava rozmerov tlačidiel – zamedzenie pretekania textu tlačidiel print č. 7 Preberanie fragmentu (Šprint 7,8,9) Posúvanie prvkov diagramu pri vkladaní nových správ Posúvanie prvkov diagramu pri vymazávaní správ	37 38 38 39 39
10 Šj 10.1 10.2 10.3 10.4	prava rozmerov tlačidiel – zamedzenie pretekania textu tlačidiel print č. 7 Preberanie fragmentu (Šprint 7,8,9) Posúvanie prvkov diagramu pri vkladaní nových správ Posúvanie prvkov diagramu pri vymazávaní správ Editovanie čiar života	37 38 38 39 39 39
0 10 Šj 10.1 10.2 10.3 10.4 10.5	prava rozmerov tlačidiel – zamedzenie pretekania textu tlačidiel print č. 7 Preberanie fragmentu (Šprint 7,8,9) Posúvanie prvkov diagramu pri vkladaní nových správ Posúvanie prvkov diagramu pri vymazávaní správ Editovanie čiar života Mazanie vrstvy ako elementu	. 37 . 38 . 38 . 39 . 39 . 39 . 39 . 41
0 10 Šj 10.1 10.2 10.3 10.4 10.5 11 Šj	prava rozmerov tlačidiel – zamedzenie pretekania textu tlačidiel print č. 7 Preberanie fragmentu (Šprint 7,8,9) Posúvanie prvkov diagramu pri vkladaní nových správ Posúvanie prvkov diagramu pri vymazávaní správ Editovanie čiar života Mazanie vrstvy ako elementu print č. 8	. 37 . 38 . 38 . 39 . 39 . 39 . 39 . 41 . 42
O 10 Šj 10.1 10.2 10.3 10.4 10.5 11 Šj 11.1	prava rozmerov tlačidiel – zamedzenie pretekania textu tlačidiel print č. 7 Preberanie fragmentu (Šprint 7,8,9) Posúvanie prvkov diagramu pri vkladaní nových správ Posúvanie prvkov diagramu pri vymazávaní správ Editovanie čiar života Mazanie vrstvy ako elementu print č. 8 Posúvanie prvkov diagramu pri vkladaní novej čiary života	. 37 . 38 . 38 . 39 . 39 . 39 . 41 . 42 . 42
O 10 Šj 10.1 10.2 10.3 10.4 10.5 11 Šj 11.1 11.2	prava rozmerov tlačidiel – zamedzenie pretekania textu tlačidiel print č. 7 Preberanie fragmentu (Šprint 7,8,9) Posúvanie prvkov diagramu pri vkladaní nových správ Posúvanie prvkov diagramu pri vymazávaní správ Editovanie čiar života Mazanie vrstvy ako elementu print č. 8 Posúvanie prvkov diagramu pri vkladaní novej čiary života Posúvanie prvkov diagramu pri vymazaní čiary života	. 37 . 38 . 38 . 39 . 39 . 39 . 39 . 41 . 42 . 42 . 42
10 Šj 10.1 10.2 10.3 10.4 10.5 11 Šj 11.1 11.2 11.3	prava rozmerov tlačidiel – zamedzenie pretekania textu tlačidiel print č. 7 Preberanie fragmentu (Šprint 7,8,9) Posúvanie prvkov diagramu pri vkladaní nových správ Posúvanie prvkov diagramu pri vymazávaní správ Editovanie čiar života Mazanie vrstvy ako elementu print č. 8 Posúvanie prvkov diagramu pri vkladaní novej čiary života Posúvanie prvkov diagramu pri vymazaní čiary života Inovované mazanie vrstiev	. 37 . 38 . 38 . 39 . 39 . 39 . 39 . 41 . 42 . 42 . 42 . 42
0 10 Šj 10.1 10.2 10.3 10.4 10.5 11 Šj 11.1 11.2 11.3 11.4	prava rozmerov tlačidiel – zamedzenie pretekania textu tlačidiel print č. 7 Preberanie fragmentu (Šprint 7,8,9) Posúvanie prvkov diagramu pri vkladaní nových správ Posúvanie prvkov diagramu pri vymazávaní správ Editovanie čiar života Mazanie vrstvy ako elementu posúvanie prvkov diagramu pri vkladaní novej čiary života Posúvanie prvkov diagramu pri vymazaní čiary života Inovované mazanie vrstiev Aktivačný blok (Šprint č. 8 a 9)	. 37 . 38 . 38 . 39 . 39 . 39 . 41 . 42 . 42 . 42 . 42 . 42
O 10 Šj 10.1 10.2 10.3 10.4 10.5 11 Šj 11.1 11.2 11.3 11.4 11.5	prava rozmerov tlačidiel – zamedzenie pretekania textu tlačidiel print č. 7 Preberanie fragmentu (Šprint 7,8,9) Posúvanie prvkov diagramu pri vkladaní nových správ Posúvanie prvkov diagramu pri vymazávaní správ Editovanie čiar života Mazanie vrstvy ako elementu posúvanie prvkov diagramu pri vkladaní novej čiary života Posúvanie prvkov diagramu pri vkladaní novej čiary života Posúvanie prvkov diagramu pri vymazaní čiary života	. 37 . 38 . 38 . 39 . 39 . 39 . 41 . 42 . 42 . 42 . 42 . 42 . 43 . 44
O 10 Šj 10.1 10.2 10.3 10.4 10.5 11 Šj 11.1 11.2 11.3 11.4 11.5 12 Šj	brava rozmerov tlačidiel – zamedzenie pretekania textu tlačidiel Preberanie fragmentu (Šprint 7,8,9) Posúvanie prvkov diagramu pri vkladaní nových správ Bosúvanie čiar života Mazanie vrstvy ako elementu brint č. 8 Posúvanie prvkov diagramu pri vkladaní novej čiary života Posúvanie prvkov diagramu pri vkladaní novej čiary života Bosúvanie prvkov diagramu pri vymazaní čiary života Bosúvanie prvkov diagramu pri vymazaní čiary života Dorava výberu elementov zo scény Bosúvanie čiar č. 9	37 38 39 39 39 39 41 42 42 42 42 42 42 43 44 45
O 10 Šj 10.1 10.2 10.3 10.4 10.5 11 Šj 11.1 11.2 11.3 11.4 11.5 12 Šj 12.1	brava rozmerov tlačidiel – zamedzenie pretekania textu tlačidiel brava výberu elementu brava výberu elementov zo scény brava výberu elementov zo scény brava výberu elementov zo scény	37 38 38 39 39 39 39 39 41 42 42 42 42 42 43 44 45
0 10 Šj 10.1 10.2 10.3 10.4 10.5 11 Šj 11.1 11.2 11.3 11.4 11.5 12 Šj 12.1 12.2	prava rozmerov tlačidiel – zamedzenie pretekania textu tlačidiel print č. 7 Preberanie fragmentu (Šprint 7,8,9) Posúvanie prvkov diagramu pri vkladaní nových správ Posúvanie prvkov diagramu pri vymazávaní správ Editovanie čiar života Mazanie vrstvy ako elementu print č. 8 Posúvanie prvkov diagramu pri vkladaní novej čiary života Posúvanie prvkov diagramu pri vymazaní čiary života Inovované mazanie vrstiev Aktivačný blok (Šprint č. 8 a 9) Oprava výberu elementov zo scény Implementácia závislostí v metamodeli Oprava RaycastCommands	37 38 38 39 39 39 39 39 41 42 42 42 42 42 43 44 45 45
O 10 Šj 10.1 10.2 10.3 10.4 10.5 11 Šj 11.1 11.2 11.3 11.4 11.5 12 Šj 12.1 12.2 12.3	prava rozmerov tlačidiel – zamedzenie pretekania textu tlačidiel print č. 7 Preberanie fragmentu (Šprint 7,8,9) Posúvanie prvkov diagramu pri vkladaní nových správ Posúvanie prvkov diagramu pri vymazávaní správ Editovanie čiar života Mazanie vrstvy ako elementu print č. 8 Posúvanie prvkov diagramu pri vkladaní novej čiary života Posúvanie prvkov diagramu pri vymazaní čiary života Inovované mazanie vrstiev Aktivačný blok (Šprint č. 8 a 9) Oprava výberu elementov zo scény print č. 9 Implementácia závislostí v metamodeli Oprava RaycastCommands Vnorený fragment	. 37 . 38 . 38 . 39 . 39 . 39 . 41 . 42 . 42 . 42 . 42 . 42 . 42 . 42 . 42
O 10 Šj 10.1 10.2 10.3 10.4 10.5 11 Šj 11.1 11.2 11.3 11.4 11.5 12 Šj 12.1 12.2 12.3 12.4	prava rozmerov tlačidiel – zamedzenie pretekania textu tlačidiel print č. 7	. 37 . 38 . 38 . 39 . 39 . 39 . 41 . 42 . 42 . 42 . 42 . 42 . 42 . 42 . 42
O 10 Šj 10.1 10.2 10.3 10.4 10.5 11 Šj 11.1 11.2 11.3 11.4 11.5 12 Šj 12.1 12.2 12.3 12.4 12.5	prava rozmerov tlačidiel – zamedzenie pretekania textu tlačidiel print č. 7 Preberanie fragmentu (Šprint 7,8,9) Posúvanie prvkov diagramu pri vkladaní nových správ Posúvanie prvkov diagramu pri vymazávaní správ Editovanie čiar života Mazanie vrstvy ako elementu porint č. 8 Posúvanie prvkov diagramu pri vkladaní novej čiary života Posúvanie prvkov diagramu pri vymazaní čiary života Inovované mazanie vrstiev Aktivačný blok (Šprint č. 8 a 9) Oprava výberu elementov zo scény Implementácia závislostí v metamodeli Oprava RaycastCommands Vnorený fragment Refactoring aplikácie	37 38 38 39 39 39 41 42 42 42 42 42 43 44 45 45 45 45 47
0 10 Šj 10.1 10.2 10.3 10.4 10.5 11 Šj 11.1 11.2 11.3 11.4 11.5 12 Šj 12.1 12.2 12.3 12.4 12.5 13 Šj	prava rozmerov tlačidiel – zamedzenie pretekania textu tlačidiel print č. 7	37 38 38 39 39 39 39 39 39 41 42 42 42 42 42 42 43 44 45 45 45 46 47 48
0 10 Šj 10.1 10.2 10.3 10.4 10.5 11 Šj 11.1 11.2 11.3 11.4 11.5 12 Šj 12.1 12.2 12.3 12.4 12.5 13 Šj 13.1	prava rozmerov tlačidiel – zamedzenie pretekania textu tlačidiel print č. 7	37 38 38 39 39 39 39 39 39 41 42 42 42 42 42 42 43 45 45 45 45 47 48
O 10 Šj 10.1 10.2 10.3 10.4 10.5 11 Šj 11.1 11.2 11.3 11.4 11.5 12 Šj 12.1 12.2 12.3 12.4 12.5 13 Šj 13.1 14 C	prava rozmerov tlačidiel – zamedzenie pretekania textu tlačidiel print č. 7	. 37 . 38 . 38 . 39 . 39 . 41 . 42 . 42 . 42 . 42 . 42 . 42 . 42 . 42
O 10 Šj 10.1 10.2 10.3 10.4 10.5 11 Šj 11.1 11.2 11.3 11.4 11.5 12 Šj 12.1 12.2 12.3 12.4 12.5 13 Šj 13.1 14 C 14.1	brava rozmerov tlačidiel – zamedzenie pretekania textu tlačidiel print č. 7 Preberanie fragmentu (Šprint 7,8,9) Posúvanie prvkov diagramu pri vkladaní nových správ Posúvanie prvkov diagramu pri vymazávaní správ Editovanie čiar života Mazanie vrstvy ako elementu posúvanie prvkov diagramu pri vkladaní novej čiary života Posúvanie prvkov diagramu pri vymazaní čiary života Posúvaníe mazanie vrstiev Aktivačný blok (Šprint č. 8 a 9) Oprava výberu elementov zo scény Porint č. 9 Implementácia závislostí v metamodeli Oprava RaycastCommands Vnorený fragment Refactoring aplikácie Testovania aplikácie Dokumentácia Pokumentácia Pokumentácia Pokumentácia Pokumentácia Pokumentácia	37 38 38 39 39 41 42 42 42 42 42 42 43 44 45 45 45 45 46 47 48 49 49
O 10 Šj 10.1 10.2 10.3 10.4 10.5 11 Šj 11.1 11.2 11.3 11.4 11.5 12 Šj 12.1 12.2 12.3 12.4 12.5 13 Šj 13.1 14 C 14.1 14.2	prava rozmerov tlačidiel – zamedzenie pretekania textu tlačidiel print č. 7	37 38 39 39 39 41 42 42 42 42 42 43 44 45 45 45 46 47 48 49 49 50

14.4	Ohraničenia produktu	
А	Používateľská príručka	A–1
A.1	Úvodná obrazovka	A–1
A.2	Práca s vrstvami	A–3
A.3	Práca s čiarami života	A–3
A.4	Práca s interakciami	A–5
A.5	Práca s fragmentom	A–7
В	Šablóna pre akceptačné testy	B–1
B.1	Čiara Života (Lifeline)	B–1
B.2	Vkladanie a zarovnávanie správ	B-4
B.3	Fragment	B–10
B.4	Mazanie Elementov	В–12
B.5	Ukladanie a Načítavanie Schémy do/z XML	B–16
С	Report z Testovania	C–1
D	Preberací protokol	D–2

1 Úvod

Tento dokument predstavuje dokumentáciu k inžinierskemu dielu tímu číslo 2: *GAMATEPI* – trojdimenzionálne UML. Obsahom tohto dokumentu je celkový pohľad na produkt vytváraný v rámci predmetu Tímový Projekt v akademickom roku 2013/2014. V kapirole č. 1 – Úvod je opísaný celkový pohľad na projekt, jeho ciele a základné implementačné informácie. V nasledujúcich kapitolách je opísaný postup prác na projekte vo forme šprintov, kde prebehla implementácia jednotlivých používateľských scenárov. V kapitole č. 8 – Celkový pohľad na projukt je opísaný celkový pohľad na projekt v kontexte softvérového inžinierstva. Kapitola sa venuje architektúre produktu a dátovému modelu. V prílohách možno nájsť používateľskú príručku a preberacie protokoly.

1.1 Celkový pohľad na projekt

UML, ako dôležitý nástroj softvérového inžiniera, slúži na návrh, špecifikáciu, modelovanie a dokumentovanie softvéru. Jednou z jeho úloh je ponúknuť grafickú alternatívu opisu softvéru. Aktuálne sa väčšina grafických nástrojov pre vizualizáciu UML zameriava len na 2D priestor. Našou snahou je vytvoriť prototyp, ktorý bude vizualizovať UML diagramy v 3D priestore.

3D zobrazenie napomáha k rýchlejšej orientácia v spleti komplikovaných vzťahov. Pridaním ďalšej dimenzie dokážeme prehľadnosť diagramu ešte zvýšiť, či zvýrazniť niektoré zaujímavé časti. Otvárajú sa tu taktiež možnosti rozšírenej vizualizácie jednotlivých častí diagramov, prípadne interakcií medzi diagramami.

Na začiatku projektu bol dodaný prototyp, ktorý obsahoval implementovanú funkcionalitu pre vytvorenie jednoduchého diagramu tried. Tento prototyp plánujeme naďalej rozvíjať. Prototyp a aj naše riešenie je zamerané na osobné počítače, avšak nie je vylúčená možnosť rozšírenia na ďalšie druhy výpočtových zariadení.

1.2 Ciele projektu

Cieľom projektu je do dodaného prototypu implementovať časti funkčnosti sekvenčného diagramu. V prvom semestri sa budeme zaoberať problematikami spustenia prototypu, vytvorenia čiary života, interakcie medzi čiarami života a základnej reprezentácie fragmentu. Samozrejmosťou je možnosť pomenovania jednotlivých prvkov diagramu, pričom nie nutne splniteľná súčasť plánu je taktiež vytvorenie vnútorného UML modelu na základe ktorého by

bolo možné vytvárať jednotlivé prvky diagramu definované v modeli. Okrem týchto funkcií spojených s diagramom je cieľom implementovať a dodať aj všeobecnú funkčnosť vytvorenia viacerých vrstiev v rôznych hĺbkach pre trojdimenzionálnosť produktu a taktiež aj serializácia vytvorených diagramov vo forme XML súborov. Cieľom je taktiež funkčnosť načítania týchto súborov a znovu vytvorenie uloženej scény.

1.3 Implementácia

Projekt nadväzuje na existujúci prototyp, ktorý je implementovaný v jazyku C++. Vo veľkom rozsahu je využívaná externá knižnica OGRE (Open Source 3D Graphic Engine). Grafické používateľské rozhranie podporuje externá knižnica MyGUI. Na kompiláciu je používaný MinGW kompilátor.

2 Ciele projektu – zimný semester

Cieľom celého projektu je vytvorenie funkčného prototypu, ktorý bude obsahovať funkčnosť vytvárania, prezerania a upravovania 3D UML sekvenčných diagramov. Táto funkčnosť bude implementovaná do existujúceho prototypu, ktorý sa ale zameriava na prácu s 3D diagramom tried. Pri ukončení práce na projekte je predpoklad zlúčenia týchto prototypov do jedného celistvého a funkčného prototypu, ktorý bude obsahovať funkčnosť oboch momentálne samostatne vyvíjaných častí prototypu.

Tretí rozmer riešenia je dosiahnutý vytvorením viacerých vrstiev, ktoré budú uložené v priestore v odlišnej hĺbke. Každá vrstva môže obsahovať celý diagram, prípadne len jeho časť. Týmto spôsobom je možné jednotlivé diagramy a ich časti jednoduchšie vizualizovať. Bude taktiež umožnená interakcia medzi vrstvami, takže samotný diagram sa nebude musieť nutne nachádzať len na jednej vrstve, ale pre zlepšenie prehľadnosti je možné diagram rozložiť do hĺbky.

Funkčnosť vytváraná v prototype musí obsahovať okrem štandardného vytvorenia, modifikácie a prehliadania diagramov aj ostatné používateľovi viditeľné, ale aj skryté funkčnosti. Vnútri diagramu je predpoklad ku možnej implementácií dátového UML metamodelu, ktorý bude ukladať prvky v diagrame a na základe neho bude môcť vytvárať nové prvky. Jedná sa napríklad o triedy a ich inštancie. Pri pomenúvaní prvkov bude ale vždy potrebné, aby používateľ vykonal vstup na základe ktorého je element pomenovaný.

Taktiež je dôležité, aby sa jednotlivé prvky diagramu navzájom na jednej vrstve neprekrývali a taktiež aby boli usporiadané podľa konvencií UML, pričom stále je dôležité, aby bol diagram prehľadný a nebol problém sa v ňom orientovať.

V prototype bude taktiež vytvorená možnosť uloženia diagramu a aj jeho načítanie z uloženého súboru.

3.1 Vytvorenie prezentačnej stránky

Za účelom prezentácie projektu sme vytvorili internetovú stránku. Stránku je možné si pozrieť na adrese <u>http://labss2.fiit.stuba.sk/TeamProject/2013/team02is-si/</u> a jej úvodná stránka sa nachádza na obrázku pod textom (Obr. 3.1). Počas celého priebehu projektu bude aktualizovaná o nové informácie a bude odzrkadľovať priebeh samotného projektu. Základnými časťami stránky sú:

- Tím Obsahuje fotografie členov tímu. Po kliknutí na jednotlivé fotografie stránka ponúkne stručné predstavenie dotyčnej osoby s jednoduchou možnosťou návratu k prehliadaniu stránky kliknutím na čierne pole stránky.
- O projekte Informácie ohľadom rozdelenia úloh v tíme ako aj základný opis nášho projektu.
- Galéria Obrazová galéria realizácie projektu.
- Dokumenty Nachádzajú sa tu dokumenty k projektu (t.j. zápisnice zo stretnutí, dokumentáciu riadenia a dokumentáciu k inžinierskemu dielu



Obr. 3.1: Úvod prezentačnej stránky

Pri implementácii prezentačnej stránky boli použité technológie HTML5, CSS3 a Javascript.

3.2 Analýza problémovej oblasti

Sekvenčný diagram je diagramom správania jazyka UML. Jeho využitie je hlavne v modelovaní scenárov prípadov použitia navrhovaného systému. V porovnaní s diagramom interakcií je výhodnejší pri malom počte komunikujúcich objektov a vyššom počte metód, cez ktoré komunikujú. Najvýraznejšou črtou tohto diagramu je zobrazenie času. Je na to využitá vertikálna os. Udalosti za sebou nasledujú zhora - nadol.

Medzi artefakty sekvenčného diagramu patrí čiara života, symbolizujúca aktéra scenára (interagujúci objekt); správy, zachytávajúce metódy a volania funkcii, a fragmenty. Fragmenty modelujú špeciálne situácie, vetvenia, cyklenia, paralelného spracovania a iných.

3.3 Spustenie existujúceho prototypu

Náš projekt sa opiera o už existujúce riešenie. To znamená, že každý člen tímu musel dokázať spustiť tento prototyp u seba lokálne. Ako bolo spomínané v kapitole 1.3 systém je založený na rozsiahlych externých knižniciach a využíva špecifický druh kompilátora, preto sa táto úloha ukázala ako netriviálna.

3.4 Príprava aplikácie

Pred začiatkom nášho vývoja bolo nutné zabezpečiť zachovanie možnosti pracovať s diagramom tried. Za týmto účelom bola v hlavnej triede zdrojového kódu vytvorená statická premenná, ktorá nadobúda hodnoty pravda/nepravda. Na základe tejto hodnoty sa na začiatku behu program určí, či bude používateľ pracovať s diagramom tried (hodnota "pravda") alebo sekvenčným diagramom (hodnota "nepravda").

V tomto štádiu nám toto riešenie prinieslo spôsob ako sa pri implementácii sústrediť na našu úlohu. Do budúcna bude možné takýto prepínač ponúknuť aj v rámci používateľského rozhrania, bez nutnosti veľkého zásahu do architektúry systému.

Pôvodný prototyp po spustení aplikácie používateľovi ponúkol na obrazovke 3 vrstvy so vzorovými triedami a vzťahmi medzi nimi. Príprava aplikácie na našu prácu preto zahŕňala aj odstránenie prebytočných prvkov scény. Výsledok tejto práce zobrazuje Obr. 3.2.





Obr. 3.2: Vľavo je stav po spustení diagramu tried. V pravo po spustení sekvenčného diagramu

3.5 Pridať čiaru života

V aplikácii bola implementovaná nová funkcionalita. Používateľ môže pridať novú čiaru života na vrstvu. Jej grafická reprezentácia pozostáva z modrého obdĺžnika, z ktorého vychádza prerušovaná čiara (viď Obr. 3.3).

Add Lifeline



Obr. 3.3: Obrazovka aplikácie po pridaní dvoch čiar života

4.1 Pridať novú vrstvu

Trojrozmernosť sekvenčného diagramu v aplikácii spočíva v jeho rozdelení na niekoľko vrstiev v priestore. Preto používateľ potrebuje možnosť pridávať nové vrstvy na obrazovku. V tomto prípade však nešlo o implementáciu novej funkcionality. Pridávanie novej vrstvy bolo už implementované v časti s diagramom tried. Práca preto spočívala v jej sprístupnení aj pre sekvenčný diagram. Výsledok po pridaní troch nových vrstiev zobrazuje Obr. 4.1.



Obr. 4.1: K jednej vrstve, vytvorenej pri spúšťaní aplikácie, sme pridali 3 nové vrstvy

4.2 Zadať názov inštancie a názov triedy pre čiaru života

Čiara života v sekvenčnom diagrame predstavuje konkrétnu inštanciu triedy (v špeciálnych prípadoch sa uvádza iba názov triedy). Tieto dva názvy sú od seba oddelené znakom ":" (dvojbodka) a zobrazujú sa v obdĺžniku čiary života.

Na základe tejto UML notifikácie bolo implementované zadávanie a zobrazovanie názvu. Používateľ určuje názov inštancie a triedy počas pridávanie novej čiary života. Obr. 4.2 zobrazuje Lifeline Creation

formulár na zadávanie textu aj so vzorkou, a na obrázku pod ním (Obr. 4.3) je výsledné zobrazenie.

Obr. 4.2: na zadanie názvu inštancie a názvu triedy

Add LifeIne Add Layer	buyersAccount :	CheckingAccount	
	ledger : Account Ledger	: Bank	

Obr. 4.3: Stav po pridaní troch čiar života

4.3 Zjednodušenie výberu čiary života

Pre zjednodušenie interakcie používateľa s prototypom bol vytvorený všeobecný obal pre grafické prvky. Obal je reprezentovaný neviditeľným obdĺžnikom, ktorý pokrýva celú plochu grafického prvku. Hlavnou úlohou obalu je zjednodušenie výberu grafického prvku používateľovi a to tak, že je postačujúce kliknúť na obal prvku. Toto môže zjednodušiť napríklad pridávanie interakcie, kedy používateľ nemusí kliknúť priamo na zvislú prerušovanú čiaru, ale stačí aby klikol do jej okolia a to v rámci šírky hlavičky čiary života.

Obal prvku, ktorý sme vytvorili, taktiež napomáha pri usporadúvaní prvkov v priestore tak, aby sa neprekrývali. Toto správanie však nie je implicitne zabezpečované obalom prvku. Jedná sa len o podporný prostriedok, ktorý si vyžaduje špecifickú implementáciu, pre každý jeden grafický prvok.

4.4 Pridať interakciu medzi objektmi

Spolupráca medzi objektmi sekvenčného diagramu je určená interakciami. Ako prvá interakcia bola implementovaná asynchrónna správa s tým, že vždy má svoj začiatok a koniec.

Ako možno vidieť na Obr. 4.4 interakcia medzi objektmi nie je obmedzená na objekty rovnakej vrstvy. Aktuálne nie sú implementované ohraničenia toho, ako pridávať interakciu, čo sa týka smeru a orientácie.



Obr. 4.4: Interakcie

4.5 Zarovnanie čiar života

Ďalším cieľom šprintu bolo zarovnanie čiar života pri ich vytvorení tak, aby sa neprekrývali a diagram bol teda prehľadný. Pri implementácií sa zistili určité problémy návrhu aplikácie, ktoré by boli ale príliš časovo náročné na opravu. Taktiež bolo navrhnuté doimplementovanie určitej funkcionality. Zarovnanie bolo vyriešené. Pri pridaní novej čiary života, sa čiara života pridá na príslušné miesto pokiaľ je tam dostatok voľného priestoru. Pokiaľ dostatok priestoru nie je, nič sa nepridá. Je to z toho dôvodu, že to by sa už jednalo o modifikáciu existujúcich čiar života, čo nie je cieľom tejto úlohy.

4.6 Uložiť a načítať diagram

Jedným z cieľov šprintu číslo 2 bola aj implementácia funkcionality na ukladanie a spätné načítavanie používateľom vytvoreného sekvenčného diagramu. Prvotný pokus o implementáciu sa opieral o princípy serializácie. Z technických príčin však vyplynulo, že ísť cestou serializácie

by si vyžadovalo zmeny v prototype v neprípustnej miere. Ako ďalšiu alternatívu sme zvolili jednoduché ukladanie dôležitých dát elementov scény, ktoré sú potrebné na spätnú rekonštrukciu diagramu v textovom súbore. Implementovaná bola základná kostra ukladania. Ďalšie práce a ukončenie ukladania a načítavania sekvenčného diagramu budú vykonané v nasledujúcom šprinte. Nižšie (Obr. 4.5) možno vidieť súčasný stav tejto funkcionality.



Obr. 4.5: Zobrazenie tlačidiel na ukladanie a načítanie sekvenčného diagramu

4.7 Oprava chýb

Po prvom šprinte obsahoval prototyp 2 zobrazovacie chyby. Obe sa nám podarilo v rámci šprintu č.2 opraviť.

Tieň v obdĺžniku čiary života

Pri prezeraní diagramu z blízka používateľ videl modrú textúru zobrazovaného obdĺžnika (reprezentujúci začiatok čiary života). Po oddialení v obdĺžniku vznikal tieň, ktorý spôsoboval esteticky nepriaznivý efekt.

Túto chybu sa nám podarilo opraviť v rámci implementácie zadávania textu do obdĺžnika (kapitola 4.2). Do obdĺžnika bol vložené textové pole, v ktorom sa tieň nevytvára.

Prerušovaná čiara presahuje vrstvu

Pri vytváraní čiary života dochádzalo ku chybe spôsobujúcej presahovanie čiary života cez spodnú časť vrstvy pre vykresľovanie. Chyba bola spôsobená fixnou dĺžkou čiary života.

Táto chyba bola opravená v rámci samostatnej úlohy, kedy bolo potrebné implementovať variabilnú dĺžku čiary života. Táto dĺžka sa vypočítava na základe pozície čiary života na vrstve a celkovej dĺžky vrstvy.

5.1 Vymazávanie objektov

Diagramy nie sú len o pridávaní nových prvkov, ale aj o ich odoberaní. V aplikácií pribudla nová funkcionalita. V menu vľavo pribudlo nové tlačidlo DeleteLifeline (Obr. 5.1). Používateľ si vyberie vrstvu, na ktorej sa čiara života nachádza pomocou oranžového obdĺžnika v pravom hornom rohu vrstvy. Po výbere vrstvy sa ostatné vrstvy skryjú. Nakoniec používateľ vyberie čiaru života, ktorú chce vymazať z diagramu. Funkcionalita vymazania čiary života zatiaľ nie je implementovaná. Pokračuje sa na nej v nasledujúcom šprinte.



Obr. 5.1: Zobrazenie tlačidla DeleteLifeline

5.2 Maximálna šírka čiary života

Pre možnosti zarovnania čiar života pri ich vytvorení je potrebné, aby bolo známe, aká môže byť maximálna šírka novovytvorenej čiary života. Táto maximálna šírka je daná konštantou vnútri

zdrojového kódu, pričom v budúcnosti je možné ju nastaviť ako súčasť konfiguračného súboru. Od tejto maximálnej šírky sa taktiež odvíja aj zalamovanie textu v čiare života.

Funkčnosť maximálnej šírky a prislúchajúcich obmedzení pri vykreslení bola úspešne implementovaná a otestovaná.

5.3 Pridanie atribútov pre prácu so šírkou a výškou elementov

Pre prácu s prvkami v grafickom prostredí bolo potrebné vytvoriť funkcionalitu ktorá umožní získanie aktuálnej šírky týchto elementov. Keďže elementy sú vykresľované tak, že sú zadefinované hraničné body polygónov a medzi týmito bodmi sú následne vytvorené čiary ktoré zobrazujú samotný element, je možné získať aktuálnu šírku, ale aj hĺbku a výšku z týchto hraničných bodov.

Výpočet je vytvorený veľmi jednoducho. Knižnica OGRE ponúka možnosť získania maximálneho a minimálneho vektora, ktorý obsahuje maximálne (minimálne) hodnoty ôs x, y a z, ktoré definujú hraničné body elementu. Na základe týchto vektorov bol vykonaný rozdiel maximálnej a minimálnej hodnoty pre danú os, pričom výsledok definuje aktuálnu šírku, výšku alebo hĺbku elementu.

5.4 Označenie čiar života ťahom myši (pre vloženie fragmentu)

Pri implementácií fragmentu bolo potrebné umožniť používateľovi vybrať si umiestnenie fragmentu. Pre podobnosť so systémami, zaoberajúcimi sa podobnou funkcionalitou, sme sa rozhodli implementovať túto funkčnosť ako ťah. Ťah sme si definovali, ako:

- 1. Stlačenie tlačidla myši na mieste, kde chceme aby fragment začal.
- 2. Držanie stlačeného tlačidla.
- 3. Presunutie sa na miesto, kde by mal fragment končiť.
- 4. Pustenie tlačidla myši.

Na začiatku sme si mysleli, že bude vhodné začať aj skončiť ťah na čiare života. Táto myšlienka sa však pri testovaní ukázala ako nepraktická. Preto sme sa rozhodli, že túto logiku upravíme v ďalšom šprinte. Nová logika sa bude opierať o body, na ktoré používateľ klikol a čiary života obsiahnuté medzi nimi.

5.5 Návrh fragmentu

Bol navrhnutý jednoduchý fragment (Obr. 5.2), podľa neho budú vytvorené ostatné typy fragmentov. Aby bolo možné v 3D diagrame vytvoriť fragment, bolo nutné implementovať spôsob výberu čiar života (*lifelines*) Tento spôsob je opísaný vyššie. Pri vytváraní má používateľ zatiaľ možnosť určiť text vyjadrujúci typ fragmentu. Na vykreslenie bol vytvorený kresliaci algoritmus, ktorého výsledok znázorňuje (Obr. 5.2). Algoritmus vykresľuje ľavú a pravú hranu fragmentu tak aby bola zarovnaná na určenú vzdialenosť od príslušnej čiary života (*lifeline*).



Obr. 5.2: Prvotný návrh a vykreslenie fragmentu

5.6 Zalamovanie textu v čiarach života

Pri implementáciu maximálnej šírky čiary života, je potrebné, aby text v jej názve zostal čitateľný, a teda aby sa prispôsobil tejto maximálnej šírke. Pri vytváraní textu môžu nastať dve možnosti. Pokiaľ je text kratší ako maximálna šírka lifeline, tak je možné ho vykresliť bez zmeny:



Obr. 5.3: Nie je potrebné modifikovať text

Ďalšou možnosť ale je, že šírka textu na jeden riadok bude väčšia ako maximálna šírka čiary života. Prvým krokom, ktorým je možné riešiť tento problém, je rozdeliť meno inštancie a meno triedy na samostatný riadok:



Obr. 5.4: Vyriešenie problému rozdelením mena inštancie a triedy na samostatné riadky

Stále môže ale nastať prípad, kde samostatný názov inštancie alebo riadku je dlhší ako maximálna možná šírka čiary života. V danom prípade sa rozdelí dlhšia časť názvu na riadky tým, že medzera v názve sa zamení za nový riadok:



Obr. 5.5: Zarovnanie pomocou rozdelenia názvov na riadky

Avšak stále môže nastať prípad, pri ktorom je samostatné slovo v názve triedy alebo inštancie dlhšie ako maximálna šírka. V tomto prípade je slovo rozdelené na viacero riadkov po dosiahnutí maximálnej šírky:



Obr. 5.6: Zarovnanie príliš dlhého slova

Takto sú všetky možné problémy vykreslenia textu ktoré môžu nastať pri príliš dlhom texte v čiare života vyriešené.

5.7 Zarovnávanie čiar života

V tomto šprinte bola doimplementovaná aj chýbajúca funkcionalita zarovnania čiar života a prislúchajúcich problémov ako zalamovanie textu, maximálna šírka, získanie aktuálnej šírky a okno s chybovou správou.

Maximálnu šírku bolo nutné doimplementovať z toho dôvodu, že aktuálny návrh aplikácie získaný od dodaného prototypu vypočítava výšku novo vytvorenej čiary života (ako aj ostatných prvkov) až pri ich vykreslení. Pre problém zarovnávania by "good practice" bolo najprv vypočítať veľkosť pridanej čiary života, následne čiaru života posunúť tak, aby bola zarovnaná, a vykresliť, avšak to možné nie je. Keďže v aktuálnom stave projektu nie je implementované upravovanie existujúcich elementov, je nutné zarovnanie čiar života vypočítavať na základe jej predpokladanej maximálnej šírky.

Používateľ pri zadávaní pozície novej čiary života musí špecifikovať jej polohu. Pokiaľ miesto ktoré označí je súčasťou existujúcej čiary života, tak pokiaľ klikol na jej ľavú polku, posunie sa stred novej čiary života na ľavo od nej, pokiaľ na pravú polku, tak na pravo od nej.

Pokiaľ používateľ klikol na aktívnu vrstvu, alebo na existujúcu čiaru života, je potrebné zistiť či v prípade vykreslenia v mieste ktoré označil nenastanú konflikty s inými existujúcimi čiarami života danej vrstvy. Tu sa kontroluje či je naľavo (napravo) od stredu novej čiary života dostatok miesta na vykreslenie. Táto kontrola prebieha na tom princípe, že v jednom pomyselnom stĺpci zadefinovanom existujúcou čiarou života, sa môže nachádzať len práve táto čiara života. Jednoduchšie povedané – čiary života sa nesmú prekrývať a nemôžu existovať jedna nad druhou. Táto kontrola sa vypočítava na základe maximálnej šírky novej čiary života a aktuálnej šírky existujúcich čiar.

Taktiež prebieha kontrola, či stred novej čiary života, ktorý bol posunutý v rámci výpočtu, sa stále nachádza na aktívnej vrstve. Pokiaľ nastane akákoľvek chyba, ktorá neumožňuje pridanie novej čiary života, prípadne pre ňu nie je dostatok miesta, tak je zobrazené chybové hlásenie a proces končí.



Obr. 5.7 Zobrazenie chybového hlásenia v prípade, že používateľ sa pokúša umiestniť novú čiaru života na pozíciu, kde nie je dostatok miesta

Funkcionalita bola implementovaná a otestovaná.

5.8 Ukladanie a načítavanie diagramov

Prvým krokom pri implementácii exportu vykresleného diagramu do XML súboru bola analýza dostupných knižníc zaoberajúcich sa práve prácou s XML v kontexte programovacieho jazyka C++. Po detailnej analýze sme vybrali knižnicu TinyXML. Integrácia do prototypu prebehla bez problémov. Ďalším krokom bola implementácia systému ukladania. Zvolili sme generický prístup, čo znamená, že v budúcnosti bude veľmi jednoduché rozšíriť alebo upraviť exportovacie schémy pre jednotlivé objekty v sekvenčných diagramoch. Úlohou pre ďalší šprint je dokončenie spätnej rekonštrukcie scény s jednotlivými elementmi na základe uloženého XML súboru.

5.9 Oprava chýb

Chybné získavanie údajov z formulára na vytvorenie čiary života

Pri vytvorení čiary života sa používateľovi zobrazí formulár, kde zadáva názov inštancie a triedy. Ak používateľ zadal dvojriadkový názov, tak aplikácia spadla. V rámci riešenia tejto chyby sme inovovali formulár, ktorý už neobsahuje iba jedno pole, ale dve (Obr. 5.8). Obidve polia sú jednoriadkové. Myslíme si, že týmto riešením sme nie len opravili existujúcu chybu, ale aj grafické používateľské rozhranie získalo na intuitívnosti.



Obr. 5.8: Formulár na zadanie atribútov čiary života

Oprava chyby výberu elementu na skrytej vrstve

Pri výbere elementu pomocou jeho obalu sa objavovala chyba, ktorá spôsobovala výber nezobrazeného elementu. Vybraný element sa nachádzal na skrytej vrstve. Pri implementácií bolo pridané overenie či sa nájdený element nachádza na aktívnej vrstve.

Nemožnosť vytvoriť čiaru života z kódu

Implementácia vykreslenia čiary života sa opierala o postupnosť funkcií vykonávanú pri vytváraní používateľom. V špeciálnych prípadoch je však potrebné vytvoriť čiaru života priamo z kódu. Preto bola logika vytvárania upravená tak, aby sa pri vykresľovaní algoritmus opieral o dáta zadané pri vytvorení čiary života – Presnejšie povedané o rodiča zadaného ako argument funkcie pri vytváraní.

6.1 Vytvorenie programovej infraštruktúry pre mazanie

Úloha "Vytvorenie stavov a tlačidla" bola upravená oproti predchádzajúcej verzií. Bolo nutné upraviť implementáciu, aby bola kompatibilná s novou verziou aplikácie. Nová logika umožňuje nielen vymazanie čiary života z diagramu, ale akéhokoľvek elementu. Bola vykonaná refaktorizácia a úpravy podľa pripomienok. Predchádzajúca verzia na zvolenej vrstve vyberala iba čiaru života a ostatné elementy ignorovala. Aktuálny stav, je taký, že je možné vybrať akýkoľvek element z vrstvy.

Spustenie mazania

Bola doplnená funkcionalita pre prepojenie výberu elementu zo scény s jeho vymazaním. Táto funkcionalita je prístupná cez tlačidlo "Delete Element", pričom je bližšie opísaná v kapitole "Vytvorenie stavov a tlačidla".

Vymazanie elementu

Bol implementovaný deštruktor, ktorý vymaže element zo zoznamu elementov, obrazovky a takisto vymaže jeho cover.

Mazanie asynchrónnych správ

Boli vytvorené deštruktory pre AsynchronousMessage, Element a Cover. Taktiež bola navrhnutá postupnosť deštruktorov elementov zobrazených na obrazovke. Každá trieda dediaca od Element musí sama zrušiť všetky referencie na seba, prípadne na ktoré sa odkazuje. Následne je zavolaný deštruktor nadtriedy (Element) v ktorom je vymazaný obal (ak nejaký má) a je vymazaná z ElementCollection a zo scény.

V rámci deštruktora asynchrónnej správy boli odstránené referencie na čiary života, ktorých sa týka. Taktiež v rámci tejto úlohy boli vytvorené metódy slúžiace na vymazanie elementu z obrazovky a z ElementCollection.

Napriek počiatočnému zadaniu tejto úlohy – vytvoreniu deštruktora, boli vytvorené metódy, ktoré umožňujú vymazania akéhokoľvek zobrazeného elementu, pričom vymazanie iných elementov bolo uľahčené a navrhnuté.

6.2 Načítavanie diagramu z XML

V tomto šprinte bola implementovaná aj funkcionalita určená na načítavanie diagramu, respektíve 3D scény so sekvenčným diagramom zo súboru vo formáte XML. Ukladanie, resp. XML reprezentácia diagramu bola implementovaná v predchádzajúcom šprinte. V tomto šprinte boli kvôli efektívnejšiemu načítavaniu vykonané zmeny aj v mechanizme ukladania. Bola pridaná aj funkcionalita pre ukladanie fragmentu. Mechanizmus načítavania je vytvorený, momentálny stav umožňuje načítať objekty typu vrstva, čiara života a asynchrónna správa. Načítanie fragmentu bude dodatočne spracované v nasledujúcom šprinte. Efektívnejšie načítavanie si momentálne vyžaduje zmeny dátového modelu, hlavne v kontexte riešenia referencií medzi jednotlivými objektami.

6.3 Označenie čiar života ťahom myši – upravená logika

Úloha označenie čiar života ťahom myši bola začatá v minulom šprinte. Avšak logika, ktorá bola pri výbere použitá, sa ukázala ako nesprávna. Preto sme logiku upravili tak, že neberieme do ohľadu to, či používateľ klikol na čiaru života. Na každej vrstve používame zoradený zoznam čiar života. Pri stlačení tlačidla myši a jeho pustení si zaznamenáme body, kde tieto udalosti nastali. Tieto 2 body nám definujú výsek, z ktorého budeme čiary života vyberať. To, či čiaru života zaradíme do výberu, záleží od jej stredového bodu (V prípade, že bod patrí do intervalu, čiaru života vyberieme).

6.4 Oprava chýb

Nepredvídateľné miznutie fragmentu

Oprava chyby, pri ktorej sa pri výbere niektorej z vrstiev nezneviditeľnili fragmenty na ostatných vrstvách.

Určenie pozície fragmentu na zadných vrstvách

Stav pre výber pozície fragmentu nenastal pri umiestňovaní na niektorú zo zadných vrstiev (Po stlačení na *layer bookmark* sa zobrazil hneď formulár). Problém bol v tom, že po stlačení na *layer bookmark* (v stave *SelectFragmentLayerState*) sa nastavil hneď nový stav (*SelectFragmentPositionState*), preto pri uvoľnení tlačidla myši sa spracovávala funkcia

processRelease už v novo nastavenom stave. Bola pridaná nová podmienka tak, aby sa funkcia *processRelease* vykonala až potom, ako sa v tom istom stave vykonala funkcia *processClick*.

Doplnenie referencií na interakcie do čiar života

Ukladanie interakcií v objekte "lifeline". Aby bolo možné vykonať zmazanie "lifeline" ako aj správ závislých na tomto objekte, bolo potrebné doplniť referencie na správy, ktoré z daného objektu "lifeline" vychádzajú aj doň vchádzajú.

7.1 Zvýraznenie obalu pri prechode cez čiaru života

V rámci používateľskej interakcie s aplikáciou bola pridaná funkcionalita na zvýrazňovanie prvkov po prechode myšou. Pri prechode myšou ponad čiaru života sa zobrazí jej obal. V prípade, že je zobrazených viac vrstiev zvýraznenie sa spustí pre všetky čiary života, ktoré pretne lúč z kurzora myši. Obr. 7.1 ukazuje zvýraznenie. Vďaka súčasnej generickej implementácii bude rozšírenie zvýraznenia pre ďalšie prvky scény triviálnou úlohou. Momentálne sa zvýrazňovanie aplikuje len pre čiaru života.



Obr. 7.1: Zvýraznenie čiary života pri prechode kurzorom. Vľavo kurzor neukazuje na čiaru života a v pravo áno.

7.2 Vymazanie aktuálneho modelu

Pre potreby importu uloženého modelu z XML súboru je potrebné vyčistiť scénu od všetkých vykreslených elementov. Preto bol doplnený deštruktor pre element typu Layer. Po zavolaní metódy ostane scéna úplne prázdna. Funkcionalitu je v budúcnosti možné rozšíriť o tlačidlo vymazania scény, aby používateľ nemusel všetko odmazávať postupne.

7.3 Optimalizácia exportu a importu scény do/z XML

V tomto šprinte prebehla aj optimalizácia mechanizmov pre export diagramu do XML a jeho načítavanie. V prípade ukladania boli do modelu pridané referencie na rodičovský element, čo nám umožnilo zredukovať množstvo ukladaných informácii (nie je naďalej nutné ukladať pre každý element zoznam jeho potomkov). V mechanizme importu scény z XML opísaná optimalizácia umožnila zredukovať používanie pomocných dátových jednotiek, čím sa redukovala spotreba operačnej pamäte a aj nároky na výpočtové prostriedky. Načítavanie prebieha hierarchicky, čo znamená, že najprv sa načítajú všetky vrstvy, následne všetky čiary života a napokon aj asynchrónne správy. Po úspešnom vytvorení dátového modelu prebehne aj proces vykreslenia diagramu do scény.

7.4 Vymazanie čiary života z diagramu

Dokončenie úlohy z predchádzajúceho šprintu. Pridaná nová funkcionalita. Používateľ môže vymazať čiaru života z diagramu ako aj všetky jej správy. Vytvorený deštruktor pre čiaru života, ktorý spustí mazanie. Obrázok vľavo znázorňuje čiary života a interakcie medzi nimi pred vymazaním a obrázok vpravo po vymazaní.

Add Lifelme Save Load Add Interaction Add Fragment DeleteElement Add Layer	instance name : Class name	Add Lifetne Bave Load dd Interaction dd Fragment AskteBerrent Add Layer	instance name	: Class name	

Obr. 7.2: Vymazanie čiary života z diagramu vľavo pred a vpravo po vykonaní operácie

7.5 Oprava chýb

Fragment sa vykresľuje ako vodorovná čiara

Pri vykresľovaní fragmentu sa stávalo, že sa vykreslila len vodorovná čiara. Táto chyba sa vyskytla v prípade, že ťah vykreslenia fragmentu bol zľava doprava. Chyba nastala pri výbere

čiar života zahrnutých do fragmentu, kedy sa hodnota druhého vybraného bodu prepísala hodnotou prvého. Táto chyba bola odstránená pridaním lokálnej premennej.

Čiara života sa zvýrazní aj keď sa nemá

V prípade, že sa zobrazila iba jedna vrstva, napríklad pri pridaní čiary života, zvýrazňovali sa aj čiary života, ktoré boli skryté. Preto bola do zvýraznenia pridaná podmienka, ktorá overuje, či je zvýrazňovaný element zobrazený.

Vykresľovanie fragmentu na prekrývajúcu sa vrstvu

Fragment sa niekedy vykreslil na vrstvu, ktorá prekrývala požadovanú vrstvu. Problém bol v tom,
že pri *raycast* funkcií sa vybrala zneviditeľnená vrstva, ktorá prekrývala požadovanú vrstvu.
Z tohto dôvodu sa aj lokálne pozície vypočítali vzhľadom na nesprávnu vrstvu, a preto sa aj fragment vykresľoval nie len na nesprávnej vrstve ale aj na nesprávnych pozíciách. Súvisiaca podmienka bola upravená tak, aby bola akceptovaná len požadovaná vrstva, nie akákoľvek.

Pri ukladaní scény program zamrzne

Ak sa ukladala scéna, ktorá obsahovala fragment s viacerými čiarami života, program sa zasekol. Problém bol v metóde *saveElement* pre uloženie fragmentu, kde sa pri ukladaní čiar života do XML súboru, program zacyklil. Zodpovedajúci cyklus bol opravený.

8 Celkový pohľad na produkt po ZS

8.1 Prototyp ako celok

Dodaný prototyp bol implementovaný v jazyku C++ za použitia grafického rámca OGRE a knižníc OIS. Na vývoj je použité prostredie Eclipse s rozšírením CDT. V projekte sme pokračovali vo vývoji s použitím doteraz používaných technológií v tom istom vývojovom prostredí. Tento prototyp bol rozšírený o dodatočnú funkcionalitu, pričom práca na ňom naďalej aktívne prebieha.

Prototyp slúži na 3D vizualizáciu UML diagramov, pričom tento projekt sa zameriava na vizualizáciu sekvenčných diagramov. Trojdimenzionálnosť produktu je dosiahnutá zobrazovaním diagramov (prípadne ich častí) v rôznych vrstvách, ktoré sa nachádzajú v rozličnej hĺbke. Pohľad na diagram je dynamický, používateľ môže prezerať jednotlivé diagramy z rôznych uhlov. Taktiež je možné sa "pohybovat" po diagrame a tým pádom sa zamerať na jednotlivé časti týchto diagramov, pričom pri pohybe po diagrame je možné pohybom priblížiť jednotlivé časti diagramu. Samotné ovládanie prototypu a aj pohybu po diagrame je veľmi jednoduché a intuitívne, pričom bližšie je opísané v používateľskej príručke, ktorá je súčasťou dokumentu.

Tento prototyp slúži na zobrazovanie a vytváranie UML sekvenčných diagramov. S použitím tretej dimenzie je možnosť posunutia diagramu do hĺbky. Priestorovosť diagramu môže byť dosiahnutá viacerými spôsobmi. Napríklad rozvrhnutím jedného diagramu do viacerých vrstiev po častiach, pričom stále je možné vytvárať interakcie medzi jednotlivými časťami diagramu napriek tomu, že sa nachádzajú na odlišných vrstvách.

Ďalším možným spôsobom je vizualizácia rôznych diagramov na odlišných vrstvách, pričom vzniká možnosť prehľadnejšieho prezerania diagramov a možných interakcií medzi nimi. Pri štandardných riešeniach by aktuálny scenár bol približne ten, že na rôznych oknách by mal používateľ vizualizované rôzne diagramy, pričom pre porovnanie týchto diagramov by bolo potrebné prepínanie medzi oknami. Avšak s pridaním tretej dimenzie do diagramu je možné toto porovnanie vykonávať v jednej obrazovke z rôznych uhlov a pohľadov.

Samotný prototyp obsahuje taktiež funkcionalitu, ktorá nie je priamo spojená s UML diagramom. Jedná sa napríklad o možnosť serializácie vytvoreného diagramu vo forme XML dokumentu pre prípad ďalšej práce na diagrame. Ďalšími riešenými problémami sú taktiež zarovnanie názvov jednotlivých čiar života, prípadne ich zarovnanie z dôvodu neprekrývania sa vo vrstvách a taktiež aj tvorba diagramov podľa špecifikácie UML.

8.2 Architektúra produktu

Prototyp je samostatnou aplikáciou, preto má iba vnútornú architektúru. Návrh je založený na šiestich balíkoch. Každý balík vyjadruje logický celok, ktorý je čo najviac oddelený od ostatných súčastí. Každý balík sa špecializuje na inú časť programovej funkcionality.

Balík OgreFramework reprezentuje pracovný rámec OGRE. Používa sa na grafické vykresľovanie všetkých objektov a správu grafického prostredia.

Balík Graphics obsahuje všetky vykresľovacie algoritmy naviazané na použitie OgreFrameworku. Algoritmy na vykreslenie sú špecifické pre jednotlivé elementy, preto je potrebné, aby každý element mal vlastný postup vykreslenia.

Balík ApplicationManagement je skupina centralizovaných tried slúžiacich na obsluhu a správu všetkých požiadaviek, ktoré vzniknú pri behu programu. Prototyp funguje ako stavový automat, ktorý pri prepínaní stavmi vykonáva iné činnosti.

Balík Core obsahuje konštruktory jednotlivých artefaktov reprezentovaných ako triedy. Konštruktory sú zapuzdrené pomocou vzoru Továreň.

Balík DataStructure je zložený z tried údajových štruktúr, ktoré reprezentujú objekty vykresľované do diagramu. Jednotlivé triedy obsahujú atribúty potrebné k serializácii.

Balík Serialization obsahuje metódy ako uložiť a načítať údajové štruktúry. Toto je potrebné pri rekonštrukcií scény a zobrazovaného diagramu.

Náčrt architektúry produktu v kontexte balíkov možno vidieť na diagrame balíkov na Obr. 8.1.



powered by Astah

Obr. 8.1: Architektúra produktu zachytená diagramom balíkov

Diagram na Obr. 8.2 znázorňuje triedu *Element* a jej súčasti. Ide o dôležitú triedu, od ktorej sú odvodené všetky kľúčové triedy súvisiace s prvkami sekvenčného UML diagramu (Obr. 8.3). Každý objekt triedy *Element* má priradený objekt triedy *ElementData* a objekt triedy *ElementGraphics*. Objekt triedy *ElementData* obsahuje polia, ktoré reprezentujú vlastnosti prvkov UML diagramu vyplývajúce zo špecifikácie UML. Objekt triedy *ElementGraphics* obsahuje atribúty grafického zobrazenia daného prvku v 3D diagrame, a tiež objekt triedy *DrawingAlgorithm*, obsahujúci algoritmus na vykreslenie daného prvku v 3D diagrame.







Obr. 8.3: Triedy odvodené od triedy Element

8.3 Dátový model

Logický model údajov je vyjadrený pomocou diagramu tried (Obr. 8.4). Popisuje hlavné prvky aplikácie. Tieto prvky predstavujú vizuálne prvky 3D diagramu. Znázornené atribúty sú použité pri ukladaní do súboru.



Obr. 8.4: Dátový model produktu zachytený diagramom tried

Layer (Vrstva)

Tento prvok predstavuje jednu vrstvu v 3D diagrame. Rôzne vrstvy sa líšia hlavne v pozícií (*z*) v rámci diagramu. *Layer* agreguje v sebe viaceré inštancie *Lifeline*, ako v klasickom sekvenčnom diagrame.

Lifeline (Čiara života)

Predstavuje jeden zo stavebných prvkov UML sekvenčného diagramu, obsahuje atribúty ako *classname* a *instancename* charakterizujúce triedu a objekt, pre ktorý bola vytvorená. Má presne určenú pozíciu v rámci *Layer* a tiež rozmery ako výška a šírka.

Asynchronous Message (Asynchrónna správa)

Predstavuje jeden zo stavebných prvkov UML sekvenčného diagramu. Pretože správa existuje medzi volajúcim a volaným objektom, v diagrame má *Asynchronous Message* určenú zdrojovú a cieľovú *Lifeline*. Atribút *methodName* predstavuje názov volanej metódy v cieľovom objekte. *SourceVertical* a *targetVertical* sú pozície začiatku a konca správy vzhľadom na príslušné *Lifeline*.

Fragment

Predstavuje ďalší prvok UML diagramu. Fragment vymedzuje určitú oblasť v rámci každej *lifeline*, ktorá doň bola zahrnutá. Fragmenty môžu vymedzovať oblasti aj v rámci iných fragmentov (Obr. 8.4).
9.1 Dokončovacie práce na funkcionalite vymazávania

Po Zimnom semestri bolo potrebné dokončiť funkcionalitu vymazávania. Problémom boli memory leaky, ktoré viedli ku spadnutiu programu a taktiež aj funkcionalita vymazania celej scény. Memory leaky boli opravené, problém tkvel hlavne v pôvodnom architektonickom návrhu, ktorý nerátal s možnosťou vymazávania objektov. Tento návrh bol pozmenený a zmena bola implementovaná, pričom tento zásah bol odkonzultovaný aj s autorom prototypu. S touto zmenou bolo vymazávanie funkčné a program bol stabilný.

Dalšou zmenou bola implementácia funkcionality vymazania scény. Táto funkcionalita je prístupná pomocou stlačenia tlačidla "Clear Scene". Po stlačení tohto tlačidla sa vymažú všetky elementy viditeľné na scéne vrátane všetkých vrstiev. Keďže by po tejto akcií na scéne nič nezostalo, tak po stlačení tlačidla sa na scéne vytvorí nová čistá vrstva.

9.2 Zarovnanie správ pri vkladaní

Keďže správy boli do diagramu umiestňované na pozície, ktoré vybral kliknutím myši používateľ, umiestňovali sa nepravidelne a tiež v rôznych uhloch, čo spôsobovalo neprehľadnosť diagramu. Vkladanie správ bolo teda upravené tak, že nové vložené správy sú kolmé na čiary života. Pri vložení sa medzi nimi vytvára fixné odsadenie. Nová správa sa vždy umiestni na koniec diagramu, na najvyššiu možnú pozíciu tak, aby nepresahovala iné správy. Na zarovnávanie bola použitá trieda *GridLayout* použitá aj pri zarovnávaní čiar života. Aby bola zachovaná prehľadnosť diagramu, a aby sa žiadne správy nekrižovali, bolo zavedené obmedzenie: žiadne dve správy nemajú spoločnú vertikálnu pozíciu. Zarovnávanie sa vykonáva v procese výberu zdrojovej a cieľovej čiary života pre správu. Ako informácia o pozícií, na základe ktorej sa správy usporadúvajú slúži aktuálna pozícia správy v scéne. Bolo doplnené tiež odsadenie správ od boxu čiary života. Zarovnanie funguje rovnako aj pre čiary života na rôznych vrstvách (Obr. 9.1).



Obr. 9.1: Príklad zarovnaných čiar života a správ medzi nimi

9.3 Zarovnanie čiar života

Čiary života sú po pridaní na vrstvu zarovnávané do mriežky. Používateľ si klikom vyberie, na ktoré miesto chce umiestniť novú čiaru života. Tá je následne automaticky zarovnaná tak, aby sa jej stred nachádzal v strede najbližšej bunky mriežky. Týmto sme dosiahly, že jednotlivé čiary života sú zarovnané do stĺpcov a do riadkov. Toto zarovnanie zároveň zabezpečuje aj minimálne rozostupy medzi čiarami života (t.j. aby sa okraj jednej nedotýkal okraja druhej). Implementovaný bol aj rám na okraji každej vrstvy. Slúži na to, aby sa čiara života nenachádzala priamo na okraji vrstvy. Na Obr. 9.2 vidíme viacero čiar života na jednej vrstve a ich zarovnanie.



Obr. 9.2: Zarovnanie štyroch čiar života na vrstve

Pre bežného používateľa je typické čítanie zľava doprava. Z tohto dôvodu sa čiary života automaticky zarovnávajú na ľavý okraj. Na Obr. 9.3 červený krížik znázorňuje miesto kde klikol používateľ, že chce umiestniť druhú čiaru života.



Obr. 9.3: Zarovnanie čiary života vľavo

9.4 Oprava chýb

Mazanie vrstvy ako elementu

Program neumožňoval vymazať samostatnú vrstvu, len prvky na nej, prípadne celú scénu. Pri riešení bola najprv zmenená funkcionalita RaycastCommad-u DeleteElement. Doteraz jej kontrola vrátila "true" pokiaľ našla nejaký element, ale pokiaľ našla Layer tak návratová hodnota bola "false". Funkcia bola zmenená tak, že ak sa nájde akýkoľvek element tak vrátila "true" (Layer dedí od Element).

Takto bolo možné vymazať celú Layer (jej deštruktor sa postaral o to, že prvky ktoré boli v nej boli vymazané). Avšak vznikali by defekty, pretože Layer neobsahuje LayerBookmark ktorá na ňu ukazuje. Preto bola do deštruktora Layer pridaná funkcionalita ktorá vyhľadala jej LayerBookmark a odmazala ho, pričom aby sa zabránilo zacykleniu pri volaní deštruktorov, tak deštruktor LayerBookmark už nevolal deštruktor Layer.

Oprava umiestnenia LayerBookmark po vymazaní scény

Po vymazaní scény a následnej práci s vrstvami si používateľ mohol všimnúť, že bookmark vrstiev sa nachádzal mimo štandardnej pozície, teda v pravom hornom rohu vrstvy. Posunutie smerom nadol vzniklo nedopatrením a bolo odstránené.

Oprava rozmerov tlačidiel – zamedzenie pretekania textu tlačidiel

Pri niektorých tlačidlách používateľského rozhrania sa vyskytovalo pretečenie textu cez hranice tlačidla, čo pôsobilo rušivo. Tento problém bol odstránený zväčšením šírky tlačidiel.

10.1 Preberanie fragmentu (Šprint 7,8,9)

Naša implementácia fragmentu sa veľmi líšila od implementácie fragmentu vytvorenej diplomantom Matejom Škodom. Preto sme sa rozhodli našu implementáciu čo najviac priblížiť jeho implementácií. Po analýze jeho verzií v gite sme prišli na to, že nemôžeme jednoducho prebrať jeho implementáciu, ale musíme navrhnúť a implementovať novú štruktúru tried slúžiacu na vytváranie fragmentu. V tejto štruktúre následne prepoužijeme veľkú čas jeho zdrojových kódov a pokúsime sa ich prispôsobiť naším potrebám. Na Obr. 10.1 je možné vidieť nami vykonanú zmenu, kedy sme pridali triedu *Container* nad triedu *Layer*. Pod triedu *Container* sme pridali ďalšiu triedu *General Fragment* a od triedy *General Fragment* trieda *Squence Fragment*.



Obr. 10.1: Nový návrh pre vytvorenie fragmentu

Takýto spôsob dedenia nám umožnil definovať základné prvky elementov, ktoré môžu obsahovať iné elementy v triede *Container*. Do triedy *General Fragment* sme prebrali značnú časť implementácie fragmentu diplomanta Mateja Škodu. Jeho implementáciu sme však najskôr museli rozdeliť na viacero menších metód. Trieda *Sequence Fragment* obsahovala našu pôvodnú implementáciu fragmentu. Logiku tejto implementácie sme ponechali, ale prebrali sme vizuálne časti implementované v triede *General Fragment*.Práce na preberaní a implementácii fragmentu prebiehali až počas 3 šprintov letného semestra.

10.2 Posúvanie prvkov diagramu pri vkladaní nových správ

Pri vkladaní nových správ má mať používateľ možnosť vkladať nové správy aj medzi existujúce správy. V tomto prípade je potrebné diagram preusporiadať tak, aby sa žiadne elementy diagramu neprekrývali a nekrižovali. To zahŕňa okrem správ aj čiary života, ktoré sa musia tiež vertikálne posúvať v prípade, že nie sú vytvorené na prvom riadku (v aktuálnej verzií neposkytujeme správy typu *create*, no umožňujeme čiary života umiestňovať aj na nižšie pozície, v budúcnosti to bude možné využiť). Špeciálnym prípadom je, keď používateľ zadá rôzne vertikálne pozície pre zdrojovú a cieľovú čiaru života. Jednak tak môže vzniknúť prekríženie s už existujúcou správou (Obr. 10.2, pozície vyznačené krúžkom), a jednak môže byť vybraná cieľová pozícia vyššia než maximálna možná zdrojová (Obr. 10.2, pozície označené štvorcom). Preto je v druhom prípade uvažovaná ako výsledná pozícia zdrojová pozícia. V prvom prípade, a aj vo všetkých ostatných je ako výsledná pozícia uvažovaná cieľová pozícia zdaná používateľom.



Obr. 10.2: Špeciálne prípady pri vkladaní interakcií

10.3 Posúvanie prvkov diagramu pri vymazávaní správ

Diagram sa musí podobne ako pri vkladaní nových správ preusporiadať aj v prípade vymazania správy. Bola implementovaná funkcionalita posúvania správ, ale aj čiar života (ak neboli na prvom riadku) pri vymazaní správy.

10.4 Editovanie čiar života

V rámci editovania čiar života sa implementovalo v tomto šprinte autoamtické editovanie. Konkrétne pre 2 situácia: vloženie čiary života medzi dve existujúce a zmazanie čiary života.

Pri vkladaní novej čiary života je potrebné aby sa existujúce čiary života rozostúpili. Na Obr. 10.3 môžeme v hornej časti vidieť stav pred vložením a v dolnej po vložení. Na hornom obrázku

červený krížik znázorňuje miesto, kde používateľ kliknutím označil, že chce vložiť novú čiaru života. Čiary života s označením 1 a 2 sa rozostúpili a medzi nimi leží čiara života 3.



Obr. 10.3: Vloženie čiary života medzi dve existujúce čiary života

Po zmazaní čiary života, od ktorej napravo sa ešte nachádzali ďalšie, ostávala na vrstve prázdna medzera. Z tohto dôvodu bolo automatické editovanie implementované aj pre mazanie čiar života. Ako vidíme na Obr. 10.4, po odstránení čiary života s označením 1 ostali na vrstve iba čiary života 3 a 2. Obe sa zarovnali ku ľavému okraju.

1 : x00000000000000000000000000000000000	3 : xxxxxxxxxxxxxxxx	2 : x00000000000000000000000000000000000	
3 : x00000000000	2 : xxxxxxxxxxxxxxxxxxxxxxxx		

Obr. 10.4: Zmena umiestnenia po zmazaní čiary života

10.5 Mazanie vrstvy ako elementu

Problémom bolo nesprávne vymazávanie vrstiev. Funkcionalita vymazávania vrstiev nebola implementovaná, ale bolo možné sa dostať do stavu v ktorom sa očakáva vymazanie vrstvy, ale nie je to programom umožnené. V prípade, keď bolo vytvorených viacero vrstiev a na žiadnej nebol iný element a používateľ chcel nejakú vrstvu vymazať sa program zasekol a nebolo možné nič vymazať, len sa vrátiť z tohto stavu stlačením tlačidla Escape. Tento problém bol vyriešený tým, že bolo umožnené vymazanie vrstvy.

11.1 Posúvanie prvkov diagramu pri vkladaní novej čiary života

Pri vkladaní novej čiary života (ktorá nie je na prvom riadku), sú prvky diagramu, ktoré sa na danej vertikálnej úrovni nachádzajú, odsadené. Bolo upravené aj vertikálne zarovnanie čiar života, ktoré sa teraz usporadúvajú na najvyššiu možnú pozíciu.

11.2 Posúvanie prvkov diagramu pri vymazaní čiary života

Ak je vymazaná čiara života, ktorá má na sebe naviazané správy, diagram sa preusporiada tak, aby boli odstránené vzniknuté horizontálne medzery (po vymazaní čiary života) aj vertikálne medzery (po odstránení správ vymazanej čiary života).

11.3 Inovované mazanie vrstiev

Doteraz bolo možné vymazať len elementy ktoré sa nachádzali na vrstvách. Bolo by veľmi výhodné keby existovala možnosť vymazania celej vrstvy spolu s jej elementami. Práve táto funkcionalita bola implementovaná. Pri vymazaní vrstvy sa vymažú aj všetky elementy na nej a aj všetky interakcie ktoré začínali/končili na vymazanej vrstve.

Pri vymazávaní vrstvi sa v prvom kroku klikne na tlačidlo "DeleteElement", označí sa vrstva na vymazanie a znova sa klikne na záložku vrstvy na vymazanie. Pre zvýraznenie toho, že je možné vymazať vrstvu a aby používateľ nebol zmätený, že musí na jedno miesto klikať dva krát, sa po označení vrstvy na vymazanie zmení farba jej záložky na červenú a tak používateľ bude vedieť, že môže vymazať celú vrstvu.



Obr. 11.1: Zvýraznenie záložky vrstvy pri vymazaní

11.4 Aktivačný blok (Šprint č. 8 a 9)

Ďalším krokom v implementácií sekvenčných diagramov bola implementácia aktivačných blokov. Postupnými iteráciami bolo implementované vytváranie aktivačných blokov po vytvorení asynchrónnej správy. Ďalším plánom bolo vytvoriť vnáranie týchto aktivačných blokov pri ich vytvorení a taktiež aj vytvorenie funkcionality predlžovania aktivačných blokov. Táto funkcionalita by našla využitie v prípade, že by používateľ chcel vyvolávať z jedného aktivačného bloku viacero správ. Avšak pokračovanie na implementácií tejto časti funkcionality bolo pozastavené. Bolo to z dôvodu presunutia ľudských zdrojov na iné dôležitejšie úlohy a taktiež aj z dôvodu veľkej časovej náročnosti. Prvotná časť implementácie tejto úlohy nie je prítomná vo výstupnom produkte, avšak všetky zdrojové kódy sú uložené vo vlastnej vývojovej vetve a pripravené pre ďalšie použitie.Príklad aktivačného bloku možno vidieť na Obr. 11.2.

Tím 2: GAMATEPI



Obr. 11.2: Príklad rozpracovaného aktivačného bloku

11.5 Oprava výberu elementov zo scény

Pri označovaní a výbere objektov zo scény bolo nemožné vymazať niektoré elementy z dôvodu ich prekrývania. Raycastovanie a následné prechádzanie cez elementy nachádzajúcej sa na rovnakej hĺbke je kvázi-náhodné (je závislé na čase pridania objektu). Bolo to vyriešené posunom elementu Message o jeden pixel dopredu pred vrstvu a posunutím elementu Fragment o dva pixely dopredu pred vrstu. Takýto grafický prvok umožní rozlíšiť vymazávanie elementov pri bežnom používaní. Ako budúce rozšírenie je možné úplne zmeniť algoritmus prechádzania raycastovaných objektov. Tento krok je však potrebné konzultovať so všetkými prispievateľmi do prototypu, pretože sa jedná o jeden zo základných pilierov celej aplikácie.

12.1 Implementácia závislostí v metamodeli

Po navrhnutí štruktúry metamodelu a jej vytvorení v rámci nášho prototypu bolo potrebné vytvoriť závislosti pri vytváraní niektorých prvkov modelu. Závislosti jednotlivých prvkov od iných sú implementované v rámci konštruktorov. Základná myšlienka bola, že pri vytvorení objektu sa tento objekt automaticky pridá do všetkých objektov, ktoré majú mať informáciu o jeho existencii. Taktiež sa vytvoria všetky elementy, ktoré vytváraný objekt potrebuje. Implementácia závislostí v metamodeli však začala až v poslednom šprinte a preto sa nám nepodarilo implementovať všetky závislosti. Implementované boli konštruktory a deštruktory pre triedy: *Space, NamedElementInLayer, Layer, LayeredNamedElement, NamedElement, InteractionFragmetn* a *Interaction*.

12.2 Oprava RaycastCommands

V prototype sa vyskytovala chyba – pri kliknutí na prázdne miesto pri výbere vrstvy(miesto kde nebol žiaden element) program spadol. Ďalšou chybou bol problém s umiestnením asynchrónnych správ. Pri existencií viacerých vrstiev za sebou sa správy vykresľovali na prvá vrstvu, nie na tú ktorá bola vybratá. Oba problémy boli opravené, výsledkom čoho je správne vkladanie asynchrónnych správ a väčšia stabilita prototypu.

12.3 Vnorený fragment

Vnorený fragment sa odstupuje od nadradeného fragmentu o dva pixely odpredu podobne ako je tomu v prípade vrstvy a fragmentu. Bolo doimplementované prepájanie podradeného fragmentu a jeho rodiča na dátovej úrovni hlavne vďaka oprave výberu elementov zo scény v predchádzajúcom šprinte. Príklad vnoreného fragmentu možno vidieť na Obr. 12.1.



Obr. 12.1: Príklad vnoreného fragmentu

12.4 Refactoring aplikácie

K vykonaniu kompletného refactoringu sa tím odhodlával dlhší čas, pretože pôvodná architektúra jednoducho nepočítala s niektorými funkcionalitami.



Obr. 12.2: Architektúra prototypu pred refactoringom

Z toho dôvodu bola ich implementácia vzhľadom na prinesenú hodnotu neúmerne dlhá a zložitá. Ku kompletnému refactoringu sa tím odhodlal v čase, keď došlo aj k implementácii metamodelu. Z pôvodnej architektúry, ktorú možno vidieť na Obr. 12.2, sme refactoringom docielili architektúru založenú na MVC architektonickom štýle, viď Obr. 12.3.



Obr. 12.3: Architektúra prototypu po refactoringu

Refactorovaný prototyp sa momentálne nachádza v samostatnej vývojovej branchi a k spojeniu s hlavnou vývojovou vetvou napriek hotovosti refactoringu nedošlo z viacerých dôvodov. Jednak nám pôsobilo veľké problémy pracovať s novou verziou vývojové prostredie Eclipse a taktiež hlavný prototyp diplomanta, ktorý pracuje s diagramami tried a aktivačnými diagramami, nepodliehal refactoringu a teda vrámci zachovania kompatibility sa vykonaný refactoring ponecháva ako návrh na ďalších pokračujúcich vývojárov.

12.5 Testovania aplikácie

Zatiaľ čo v predchádzajúcom šprinte tím vypracoval sadu akceptačných testov, v tomto šprinte prebehlo samotné testovanie. Report a kompletné informácie z testovania možno vidieť v kapitole *Report z Testovania*.

13.1 Dokumentácia

V poslednom týždňovom šprinte boli všetky sily členov tímu smerované k tvorbe dokumentácie k finálnemu produktu, čoho výsledkom je aj tento dokument. Taktiež sme opravovali chyby, ktoré boli výstupom testovania v predchádzajúcom šprinte.

14 Celkový pohľad na produkt

14.1 Výsledky

Pomocou nami vytvoreného prototypu sme mali zistiť, či je možné modelovanie sekvenčného diagramu v 3D priestore. Túto úlohu sa nám podarilo splniť. Nami vytváraný sekvenčný diagram môže obsahovať čiary života, interakcie medzi správami a fragmenty. Čiary života a interakcie medzi nimi je možné pridávať a mazať, pričom diagram sa po každej takejto akcii automaticky preusporiada. Troj-dimenzionálnosť diagramov sme dosiahli modelovaním pomocou vrstiev. Všetky vrstvy spolu tvoria jeden diagram a preto je možné vytvárať interakcie aj medzi vrstvami. Ukážku diagramu je vidieť na obrázku Obr. 14.1. Takýto diagram je možné uložiť a načítať.



Obr. 14.1: Ukážka prototypu

V prototype začatá implementácia aktivačných blokov. Práca na aktivačných blokoch však musela byť pozastavená. Príčinou pozastavenia prác bol problém s modelom, ktorý bol implementovaný v prototype. Preto sme sa rozhodli tento model zmeniť na model zhodný s UML Superstructure 2.4.1. Tento model však bolo potrebné rozšíriť o 3D priestor, keďže UML je vytvorené pre 2D diagramy. Implementácia nového modelu však začala v pokročilom štádiu

projektu. Preto sme nestihli dokončiť celú implementáciu. Podarilo sa nám implementovať štruktúru a konštruktory a deštruktory niektorých objektov.

14.2 Architektúra

Architektúra prototypu ktorý bol dodaný na začiatku projektu sa výrazne líši od architektúry ktorá je aktuálne prítomná v odovzdávanom prototype. Stále má samostatnú vnútornú architektúru a je postavený na OGRE frameworku, avšak najdôležitejšou zmenou bol presun jednotlivých balíkov a tried tak, aby prototyp zodpovedal Model-View-Controller (MVC) architektúre. Boli vytvorené jednotlivé balíky ktoré zodpovedajú jednotlivým komponentom tohto MVC modelu, do ktorých boli presunuté jednotlivé balíky pôvodného prototypu. Týmto bolo zabezpečené väčšie oddelenie funkcionality. Jednotlivé rozdelenie pôvodných balíkov do MVC modelu je zobrazené na Obr. 14.2.



powered by Astah

Obr. 14.2: Architektúra prototypu založená na arch. štýle MVC

Okrem zmeny na MVC architektúru bol taktiež zrušený balík *Serialization*, pretože tento balík stratil s postupom implementácie svoj význam. Popis funkcionalít jednotlivých komponentov MVC modelu je nasledovný:

- 1. *DataStructure* obsahuje triedy ktoré reprezentujú objekty vykresľované do sekvenčného diagramu.
- 2. *Graphics* obsahuje algoritmy ktoré sú zodpovedné za vykresľovanie objektov, pričom tento komponent je silno previazaný s funkcionalitou OGRE Framework-u.
- 3. Gui obsahuje triedy ktoré reprezentujú vizuálne okná a tlačidlá na scéne.
- 4. *Core* obsahuje konštruktory jednotlivých objektov ktoré sú zobrazované v diagrame, pričom sú zapuzdrené pomocou architektonického vzoru *Factory Method*.
- 5. *ApplicationManagment* obsahuje skupinu tried slúžiacich pre obsluhu všetkých požiadaviek ktoré vzniknú počas behu programu.

Počas letného semestra bola vyvinutá práca pre vytvorenie a implementáciu dátovej časti podľa UML Superstructure 2.4.1. Bola prebratá časť ktorá slúžila pre popis sekvenčného diagramu, avšak bola modifikovaná tak, aby vyhovovala pre našu víziu použitia sekvenčných diagramov v 3D priestore. Metamodel je reprezentovaný diagramom tried na Obr. 14.3.

Tento diagram poskytuje všetky triedy ktoré sú potrebné pre plnohodnotnú reprezentáciu sekvenčného diagramu. Nasleduje popis najdôležitejších tried pre pochopenie metamodelu - *Interaction* predstavuje diagram ako celok, *Lifeline* čiaru života, *Message* asynchrónnu alebo synchrónnnu správu, pričom každá správa má svoj začiatok a koniec ktoré sú reprezentované triedou *MessageEnd* a poradie správ je reprezentované pomocou triedy *OccurenceSpecification*. Ďalej trieda *InteractionFragment* predstavuje objekt ktorý môže mať v sebe časť diagramu. Trieda *CombinedFragment* je jej spresnením pričom predstavuje fragment tak ako ho poznáme v sekvenčných diagramoch.

Štruktúra tried v aplikačnej časti zostala zachovaná podľa architektúry zimného semestra, pričom všetky objekty stále dedia od triedy *Element*.

Keďže metamodel predstavený v UML superstructure nepočíta s využitím sekvenšného diagramu v 3D priestore, bolo potrebné ho dotvoriť tak, aby mohol slúžiť účelu tohto projektu. Podľa UML Superstructure každá trieda metamodelu dedí od triedy *NamedElement*. My sme na základe tejto znalosti rozšírili metamodel o triedy zobrazené v diagrame tried na Obr. 14.4.



Obr. 14.3: Diagram tried UML metamodelu 2.4.1 sekvenčného diagramu



powered by Astah

Obr. 14.4: Diagram tried reprezentujúci rozšírenie UML metamodelu

Zachovali sme pôvodnú triedu *NamedElement*, pričom každá inštancia tejto triedy patrí do jedného 3D priestoru reprezentovaného triedou *3DSpace*. Keďže v našom prototype v 3D priestore je možné mať viacero vrstiev, tak pre reprezentáciu vrstvy sa použije trieda *Layer*, ktorá patrí vždy práve jednému 3D priestoru. V našom prototype sa môžu vyskytovať triedy ktoré patria do vrstvy, alebo patria do celého priestoru. Preto je vytvorená trieda *LayeredNamedElement*, ktorá zodpovedá takému elementu, ktorý je súčasťou nejakej vrstvy, prípadne viacerých vrstiev. Ďalej je to v prototype použité tak, že pokiaľ má byť určitý *Element* súčasťou vrstvy, tak dedí od triedy *LayeredNamedElement* a pokiaľ má byť súčasťou celého priestoru, tak dedí od triedy *NamedElement*.

Aktuálne je implementovaná štruktúra jednotlivých častí novej dátovej štruktúry (podľa UML metamodelu), avšak nie je napojená na prototyp, stále sa využíva stará štruktúra založená na návrhovom vzore *Composite* ktorá je podrobnejšie opísaná v časti zimného semestra.

14.3 Opis funkcionality

Prototyp projektu trojdimenzionálneho UML je samostatnou aplikáciou, umožňujúcou prácu so sekvenčným diagramom v 2.5D priestore, čo znamená, že aplikácia používa viacero vrstiev uložených za sebou v priestore. Samotné prvky sekvenčného diagramu su ukladané na tieto

vrstvy, vo výnimočných prípadoch medzi (napríklad interakcie). Počet vrstiev nie je limitovaný. Používateľ môže pridávať nové a mazať existujúce vrstvy podľa svojich osobných preferencií.

Používateľovi je umožnené pridávať nové čiary života. Názov objektu a triedy sa zadáva pomocou grafického formulára. Tieto informácie sa z formulára extrahujú a sú zobrazené v obdĺžniku čiary života oddelené znakom dvojbodka, tak ako to poznáme zo štandardov UML. V našom prototype sme rátali s možnosťou, že používateľ zadá príliš dlhý názov objektu alebo triedy. V takomto prípade sa zväčší výška obdĺžnika čiary života a názov je rozdelený na niekoľko riadkov pod sebou, tak aby bola zachovaná maximálna šírka čiary života a aby z nej text nepresahoval.

Umiestnenie novej čiary života je dané kombináciou vstupu používateľa a automatického zarovnania. Zarovnanie čiar života je do ortogonálnej pravidelnej mriežky. Po okrajoch vrstvy je neviditeľný rám, kam nie je možné umiestňovať žiadne prvky sekvenčného diagramu. Ak sa používateľ pokúsi pridať čiaru života do tohto rámu je jej pozícia prerátaná do najbližšej bunky mriežky. Spodný rám zase predstavuje miesto kam už prerušovaná čiara čiary života nezasahuje a na jeho okraji končí. Mriežkou teda zabezpečujeme zarovnanie čiar života do stĺpcov a do riadkov s rovnakými rozostupmi. Zároveň je zabezpečené aby sa čiary života neprekrývali ani nedotýkali. K tomuto je ešte implementovaná funkcionalita na zarovnávanie čiar života vľavo. Používateľovi je umožnené aj vkladanie novej čiary života medzi už existujúce. Následne sa pôvodné čiary života premiestnia, tak aby bolo zachované rozmiestnenie do mriežky.

Medzi ľubovoľné dve čiary života je možné umiestniť interakciu. Tieto čiary života môžu ale nemusia byť na rovnakej vrstve. Názov interakcie je opäť zadávaný používateľom. Rovnako ako pri čiarach života, aj pri interakciách je výsledné umiestnenie dané kombináciou vstupu používateľa a automatického zarovnania. Všetky interakcie sú zarovnávané zhora nadol s fixnými medzerami. Interakcie sa vykresľujú kolmo na čiary života automaticky a nie je nutné aby sa používateľ trápil s presným kliknutím pre docielenie tohto efektu. Je zabezpečené aby sa interakcie neprekrývali medzi sebou ani s obdĺžnikmi čiar života. Interakcie sa prispôsobujú pohybom čiar života spôsobenými zmenou rozmiestnenia. To znamená, že ich veľkosť a smer sa mení tak, aby vždy spájali tie dve čiary života, ktoré používateľ zadal pri ich vytváraní (aj pri zmenách ich umiestnenia).

Tretím prvkom sekvenčného diagramu, ktorý bol implementovaný, je fragment. Fragment sa v čase odovzdávania tohto projektu nachádza v stave, v ktorom je možné vytvárať ho iba nad čiarami života nachádzajúcimi sa na jednej vrstve. Tieto čiary života sa označujú ťahom myši

ponad. Pri vytvárani fragmentu zadáva používateľ typ fragmentu, ktorý sa následne zobrazí v jeho hlavičke. Vnorené fragmenty neboli implementované.

Aktivačný blok pre čiary života bol implementovaný v zjednodušenej verzii. Používateľ mohol vytvoriť aktivačný blok pre jednu interakciu. Pre nízku prioritu táto funkcionalita nebola dokončená. V súčasnosti ju prorotyp neposkytuje.

Funkcionalita mazania bola implementovaná pre nasledujúce prvky: čiara života, interakcia a vrstva. Zmazanie čiary života, ak je to potrebné, sa odzrkadlí v rozmiestnení ostatných čiar života, tak aby bolo zachované ich zarovnanie doľava. Podobný efekt dostávame aj pri zmazaní interakcie, kde zase zostávajúce interakcie zarovnáme nahor, tak aby po zmazanej neostalo v zobrazení prázdne miesto. Zamazaním čiary života sa zmažu aj všetky interakcie, ktoré do nej alebo z nej viedli. Pri zmazaní vsrtvy sa zase zmažú všetky na nej ležiace čiary života a interakcie s nimi spojené. Používateľovi je umožnené zmazať celú scénu naraz kliknutím jedného tlačidla. Po takomto vyčistení scény používateľ modeluje od začiatku z pozície s jednou prázdnou vrstvou.

Pre pohodlné ovládanie aplikáciem sme čiary života obklopili obalom. Vďaka tomuto obalu je jednoduchšie kliknúť miesto na prerušovanej čiare čiary života, napríklad pri pridávaní novej interakcie. Tento obal sa zvýrazní vždy pri prechode myšou ponad príslušnú čiaru života.

Funkcionalita ukladania modelu do XML a jeho spätného načítavania bola úspešne implementovaná v zimnom semestri. Vzhľadom na rozsiahle zmeny v modeli, ktoré boli v letnom semestri vykonané, sa túto funkcionalitu nepodarilo zachovať.

14.4 Ohraničenia produktu

Výsledný prototyp ako nástroj na modelovanie sekvenčných UML diagramov má nasledovné ohraničenia:

- neumožňuje modelovať všetky prvky diagramu, iba ohraničené verzie nasledovných prvkov: čiara života, správa, fragment
- nie je umožnené editovať prvky diagramu (pozície, textový obsah)
- zarovnanie prvkov je možné zatiaľ len pre čiary života a správy
- fragment je možné vytvárať na jednej vrstve

A Používateľská príručka

Pre spustenie aplikácie je potrebné spustiť súbor 3DUML.exe (jediný vykonateľný súbor v inštalačnom adresári). Pri prvom štarte je potrebné zvoliť vykresľovací engine, pričom odporúčame zvoliť engine "Direct3D" (Obr. príloha A.1). Túto akciu však nie je potrebné opakovať pri ďalších spusteniach programu.

OGRE Engine Rendering Setup		F
Rendering Subsystem: - Rendering System Options OpenGL Rendering Subsystem	0	
[Click On An Option]:	OK	<u>+</u> Cancel

Obr. príloha A.1: Štartovacia obrazovka pri prvom spustení aplikácie

A.1 Úvodná obrazovka

Následne Vás program presmeruje na úvodnú obrazovku (Obr. príloha A.2), kde môže používateľ začať modelovať sekvenčný diagram. Na úvodnej obrazovke vidí používateľ nasledujúce ovládacie prvky:

- 1. Tlačidlo "Add Lifeline" slúži na pridanie čiary života,
- 2. Tlačidlo "Save" slúži na export modelovaného diagramu vo formáte XML,
- Tlačidlo "Load" slúži na načítanie existujúceho modelu v súbore vo formáte XML do programu,
- 4. Tlačidlo "Add Interaction" slúži na vytvorenie interakcie medzi dvoma čiarami života,
- Tlačidlo "Add fragment" slúži na vytvorenie fragmentu z jednej alebo viacerých čiar života,
- Tlačidlo "Delete Element" slúži na vymazanie prvku sekvenčného diagramu (Čiara života, interakcie, fragment, vrstva),
- 7. Tlačidlo "Add Layer" slúži na pridanie ďalšej vrstvy do súčasného modelu,
- 8. Tlačidlo "Clear Scene" slúži na kompletné zmazanie súčasného diagramu a vytvorenie defaultnej obrazovky so základným modelom obsahujúcim jednu vrstvu,
- 9. Prvá vrstva, ktorá je základom každého vytvoreného modelu.



Obr. príloha A.2: Základná obrazovka po vstupe do programu

A.2 Práca s vrstvami

S vrstvami sa pracuje veľmi jednoducho. Pre pridanie novej vrstvy je potrebné, aby používateľ stlačil tlačidlo "Add Layer". Nová vrstva je automaticky pridaná za už existujúce vrstvy diagramu. Diagram s viacerými vrstvami možno vidieť na Obr. Príloha A.3



Obr. Príloha A.3: Diagram s viacerými vrstvami

Vymazanie vrstvy je rovnako jednoduché:

- 1. Kliknutie na tlačidlo "Delete Element"
- 2. Výber vrstvy, ktorú chce používateľ zmazať kliknutím na záložku vrstvy
- Opakované kliknutie na záložku vrstvy, ktorú používateľ vybral v kroku č. 2 (táto záložka je po výbere v kroku č. 2 zafarbena na červeno, indikujúc fakt, že ide o vrstvu, na ktorej používateľ vykonáva operáciu mazania elementu)

A.3 Práca s čiarami života

Pre vytvorenie čiary života je potrebné zvoliť tlačidlo "Add Lifeline" z ľavého ovládacieho panelu aplikácie. Následne sa pri všetkých vrstvách v diagrame zobrazí na ich pravej strane

v hornej časti oranžový štvorček (Ovládací prvok ,1' na Obr. príloha A.4). Vďaka tejto záložke je možné vybrať vrstvu, na ktorú chce používateľ pridať čiaru života (Tento postup je rovnaký pri pridávaní akéhokoľvek elementu, resp. všeobecne pri akciách vyžadujúcich výber vrstvy).



Obr. príloha A.4: Pridávanie prvku na vrstvu pomocou záložky vrstvy

Následne je používateľ vyzvaný k zadaniu základných informácií o vkladanej čiare života. Požadované informácie a spôsob ich vyplnenia vidíme na Obr. príloha A.5. Dialógové okno má nasledujúce ovládacie prvky:

- 1. Názov inštancie
- 2. Názov triedy
- Tlačidlo "Create" pre ukončenie editácie hodnôt a potvrdenie aktuálne vložených hodnôt
- 4. Tlačidlo "x" pre zrušenie akcie pridania čiary života.

Používateľská príručka



Obr. príloha A.5: Dialógové okno vyžadujúce dodatočné informácie potrebné pre vytvorenie čiary života od používateľa

Podobné dialógové okná sa nachádzajú na viacerých miestach aplikácie, avšak veríme, že ich jednoduchosť nevyžaduje dodatočnú pozornosť v tejto používateľskej príručke. Zmazanie čiary života prebieha nasledovne:

- 1. Stlačenie tlačidla "Delete Element"
- 2. Výber vrstvy s čiarou života, ktorú chceme mazať (viď. Obr. príloha A.4)
- 3. Výber čiary života a potvrdenie vymazania kliknutím na vybratú čiaru života.

A.4 Práca s interakciami

Pre vytvorenie interakcie je potrebné mať v diagrame aspoň jednu čiaru života. Interakcia má zdrojovú a cieľovú čiaru života a keďže tie sa môžu zhodovať, interakciu je možné vytvoriť na jednej čiare života, alebo medzi dvoma čiarami života, pričom interakcie je možné vytvárať aj medzi čiarami života, ktoré sa nenchádzajú na rovnakej vrstve (Obr. príloha A.6).

Používateľská príručka



Obr. príloha A.6: Diagram s interakciami medzi čiarami života na rozličných vrstvách

Postup vytvorenia interakcie:

- 1. Kliknutia na tlačidlo "Add Interaction"
- 2. Výber vrstvy so zdrojovou čiarou života kliknutím na záložku príslušnej vrstvy
- 3. Výber konkrétnej čiary života kliknutím na čiaru života na vrstve vybranej v kroku 2
- 4. Výber vrstvy s cieľovou čiarou života kliknutím na záložku príslušnej vrstvy
- 5. Výber konkrétnej čiary života kliknutím na čiaru života na vrsve vybranej v kroku 4
- Vyplnenie názvu interakcie v dialógovom okne (Obr. príloha A.7) a potvrdenie výberu kliknutím na tlačidlo "Create".

Používateľská príručka



Obr. príloha A.7: Vloženie názvu interakcie a potvrdenie vloženia

Pre vymazanie interakcie opakujte postup z kapitoly o práci s čiarou života, stačí zameniť čiaru života za interakciu:

- 1. Stlačenie tlačidla "Delete Element"
- 2. Výber vrstvy s interakciou, ktorú chce používateľ mazať (viď. Obr. príloha A.4)
- 3. Výber interakcie a potvrdenie vymazania kliknutím na vybratú interakciu

A.5 Práca s fragmentom

Pre vytvorenie fragmentu v sekvenčnom diagrame je vyžadovaná prítomnosť aspoň jednej čiary života. Fragment sa vytvára nasledujúcim spôsobom:

- 1. Používateľ klikne na tlačidlo "Add Fragment" v ľavom ovládacom menu aplikácie
- 2. Používateľ zvolí vsrtvu, na ktorej chce fragment vložiť kliknutím na záložku vrstvy
- 3. Používateľ stlačením ľavého tlačidla myši zvolí počiatočný bod fragmentu (ľavý horný roh). Nasleduje výber čiar života, resp. elementov, ktoré má fragment zahŕňať. Tento

výber používateľ vykoná ťahom myši so stlačeným ľavým tlačidlom a tlačidlo pustí až nad bodom (pravý dolný roh), kde má fragment končiť

4. Používateľ je vyzvaný v dialógovom okne vyplniť názov fragmentu. Výber a vloženie fragmentu potvrdí kliknutím na tlačidlo "Create"

Pre vymazanie fragmentu je potrebné vykonať nasledujúcu sekvenciu krokov:

- 1. Stlačenie tlačidla "Delete Element"
- 2. Výber vrstvy s fragmentom, ktorý chce používateľ mazať (viď. Obr. príloha A.4)
- 3. Výber fragmentu a potvrdenie vymazania kliknutím na vybratý fragment

B Šablóna pre akceptačné testy

V tejto kapitole uvádzame šablónu akceptačných testov. Podľa týchto testov prebehlo testovanie produktu nezaujatými tretími stranami. Výsledky tohto testovania možno vidieť v prílohe C. Presné výsledky z dôvodu rozsiahlosti neuvádzame v tomto dokumente, avšak nachádzajú sa na priloženom médiu.

B.1 Čiara Života (Lifeline)

ID	1	Názov	Vloženie prvej čiary života			
Rozhranie Sekvenčr		ý diagram – Čiara života				
Účel Overenie		funkčnosti vytv	unkčnosti vytvorenia novej čiary života			
Vstupné podmienky		Je vytvorená prázdna vrstva				
Výstupné podmienky		Čiara života je vytvorená, viditeľná na obrazovke, zarovnaná do ľava v rovnakej výške v akej sa kliklo				
Krok	Akcia	Akcia		Očakávaná reakcia	Skutočná reakcia	
1	Stlače	Stlačenie medzerníka		Zobrazí sa kurzor		
2	Kliknu lifeline	Kliknutie na tlačidlo "Add lifeline"		Zobrazia sa záložky pre všetky vrstvy, kurzor je viditeľný		
3	Výber vrstvy, na ktorú chceme umiestniť čiaru života		ktorú iť čiaru ži∨ota	Všetky ostatné vrstvy sa zneviditeľnia		
4	Kliknu miest čiaru	Kliknutie pomocou kurzora na miesto kam chceme vložiť čiaru života		Zobrazí sa dialógové okno na vyplnenie informácií o čiare života		
5	Kliknutie na tlačidlo "Create"		dlo "Create"	Vytvorí sa čiara života a je zarovnaná do ľava		

ID	2	Názov	Vloženie čiary života medzi dve existujúce čiary života				
Rozhranie Sekvenčný d			ý diagram – Čiara života				
Účel		Overenie	funkčnosti vytv	vorenia novej čiary života			
Vstupné podmienky Je vy			Je vytvorená	rtvorená vrstva, na ktorej sú 2 čiary života			
Výstupné podmienky		Čiara života je vytvorená, viditeľná na obrazovke, vložená medzi 2 na začiatku existujúce. Žiadne čiary života sa neprekývajú					
Krok	Akcia	Akcia		Očakávaná reakcia	Skutočná reakcia		
1	Stlač	Stlačenie medzerníka		Zobrazí sa kurzor			
2	2 Kliknutie na tlačidlo "Add lifeline"		dlo "Add	Zobrazia sa záložky pre všetky vrstvy, kurzor je viditeľný			
3	3 Výber vrstvy, na ktorú chceme umiestniť čiaru života		ktorú ť čiaru života	Všetky ostatné vrstvy sa zneviditeľnia			
4	Klikn medz života	Kliknutie pomocou kurzora medzi 2 existujúce čiary života		Zobrazí sa dialógové okno na vyplnenie informácií o čiare života			
5	5 Kliknutie na tlačidlo "Create"		dlo "Create"	Medzi 2 existujúce čiary života sa vložila nová čiara života			

ID	3	Názov	Zvýraznenie čiary života po prechode myšou			
Rozhranie Sekvenčný diagram –		ý diagram – Č	iara života			
Účel Overenie funkčnosti zvýraznenia			funkčnosti zvý	raznenia čiary života		
Vstupné podmienky Je vytvorena		Je vytvorená	vrstva, na ktorej je vytvorená čiara života			
Výstupné podmienky S			Stav na scén	e ostal nezmenený		
Krok	rok Akcia			Očakávaná reakcia	Skutočná reakcia	
1	Stlač	Stlačenie medzerníka		Zobrazí sa kurzor		
2 Posun kurzora tak, aby ukazoval na čiaru života		ik, aby u života	Okolo čiary života sa zobrazí čierny obdĺžnik			
3 Posun kurzora tak, aby neukazoval na čiaru života		ik, aby aru života	Čierny obdĺžnik okolo čiary života zmizne			

B.2 Vkladanie a zarovnávanie správ

ID	4	Názov	Vloženie správy medzi dve čiary života			
Rozhranie Sekvenčný diagra		ý diagram – V	gram – Vloženie správy			
Účel Overenie funkčnosti vytv			funkčnosti vytv	vorenia a zarovnania správy		
Vstupné podmienky Je vytvorená			Je vytvorená	prázdna vrstva obsahujúca dve čiary života (v rovnakej výške)		
Výstupné podmienky Správa je vy			Správa je vyt	vorená medzi čiarami života a je zarovnaná na najvyšiu možnú pozíciu		
Krok	Krok Akcia			Očakávaná reakcia	Skutočná reakcia	
1	Stlače	Stlačenie medzerníka		Zobrazí sa kurzor		
2	Kliknı Intera	Kliknutie na tlačidlo "Add Interaction"		Zobrazia sa záložky pre všetky vrstvy, kurzor je viditeľný		
3	Výber života vrstvy	Výber vrstvy zdrojovej čiary života kliknutím na záložku vrstvy		Všetky ostatné vrstvy sa zneviditeľnia		
4	Kliknı jednu	Kliknutie pomocou kurzora na jednu z čiar života		Zobrazia sa záložky pre výber vrstvy cieľovej čiary života		
5	Výber života vrstvy	Výber vrstvy cieľovej čiary života kliknutím na záložku vrstvy		Všetky ostatné vrstvy sa zneviditeľnia		
6	Kliknı ďalšiu	Kliknutie pomocou kurzora na ďalšiu z čiar života		Zobrazí sa dialógové okno na vyplnenie textu správy		
7	Kliknutie na tlačidlo "Create"		dlo "Create"	Vytvorí sa správa medzi zvolenými čiarami života a je zarovnaná na najvyššiu možnú pozíciu tak, aby neprekrývala rámik žiadnej čiary života		
ID	5	Názov	Vloženie spra	ávy medzi dve čiary života, pričom jedna je numiestnená nižšie ako druhá		
---	---	--	--	---	---------------------	--
Rozhranie Sekvenčný diagram – Vloženie správy						
Účel		Overenie	funkčnosti vytv	vorenia a zarovnania správy		
Vstupné po	dmienky		Je vytvorená	prázdna vrstva obsahujúca dve čiary života (jedna čiara života je	umiestnená nižšie)	
Výstupné p	odmienk	у	Správa je vyt	vorená medzi čiarami života a je zarovnaná na najvyšiu možnú po	zíciu	
Krok	Akcia	1		Očakávaná reakcia	Skutočná reakcia	
1	Stlače	enie medze	rníka	Zobrazí sa kurzor		
2	Kliknu Intera	utie na tlačio oction"	dlo "Add	Zobrazia sa záložky pre všetky vrstvy, kurzor je viditeľný		
3	Výbei umies kliknu	Výber vrstvy zdrojovej (nižšie umiestnenej) čiary života kliknutím na záložku vrstvy		Všetky ostatné vrstvy sa zneviditeľnia		
4	Kliknu nižšie života	Kliknutie pomocou kurzora na nižšie umiestnenú čiaru života		Zobrazia sa záložky pre výber vrstvy cieľovej čiary života		
5	Výbei umies kliknu	Výber vrstvy cieľovej (vyššie umiestnenej) čiary života kliknutím na záložku vrstvy		Všetky ostatné vrstvy sa zneviditeľnia		
6	Kliknu vyššie života pozíc vyššia pozíc čiare	utie pomoco e umiestner a, pričom zv ia v rámci č a, než najvy ia v zdrojov života	ou kurzora na nú čiaru rolená iary života je vššia možná rej (nižšej)	Zobrazí sa dialógové okno na vyplnenie textu správy		
7	Kliknu	utie na tlačio	dlo "Create"	Vytvorí sa správa medzi zvolenými čiarami života a je zarovnaná na najvyššiu možnú pozíciu tak, aby neprekrývala rámik žiadnej čiary života		

ID	6	Názov	Vloženie spra	ávy medzi dve čiary života a zároveň nad existujúcu správu		
Rozhranie Sekvenčný diagram – Vloženie správy						
Účel		Overenie	funkčnosti vytv	vorenia a zarovnania správy a ostatných správ		
Vstupné po	dmienky		Je vytvorená	prázdna vrstva obsahujúca dve čiary života, medzi ktorými je vytv	orená správa	
Výstupné p	odmienk	у	Správa je vyt nižších, ako :	tvorená medzi čiarami života, existujúce správy, ktoré boli umiestn zvolená pozícia, sú posunuté nižšie a žiadne správy sa neprekrýva	ené na pozíciách ajú	
Krok	Akcia	I		Očakávaná reakcia	Skutočná reakcia	
1	Stlače	enie medze	rníka	Zobrazí sa kurzor		
2	Kliknu Intera	utie na tlačio ction"	dlo "Add	Zobrazia sa záložky pre všetky vrstvy, kurzor je viditeľný		
3	Výbei života vrstvy	Výber vrstvy zdrojovej čiary života kliknutím na záložku vrstvy		Všetky ostatné vrstvy sa zneviditeľnia		
4	Kliknu ľubov z čiar	Kliknutie pomocou kurzora na ľubovoľnú pozíciu v jednej z čiar života		Zobrazia sa záložky pre výber vrstvy cieľovej čiary života		
5	Výbei života vrstvy	Výber vrstvy cieľovej čiary života kliknutím na záložku vrstvy		Všetky ostatné vrstvy sa zneviditeľnia		
6	Kliknu ďalšiu zvole života už ex čiare	utie pomoco i z čiar živo ná pozícia v i je vyššia r istujúcej sp života	ou kurzora na ta, pričom / rámci čiary lež pozícia rávy na tejto	Zobrazí sa dialógové okno na vyplnenie textu správy		
7	Kliknu	utie na tlačio	dlo "Create"	Vytvorí sa správa medzi zvolenými čiarami života, správa, ktorá bola umiestnená pod zvolenou pozíciou je posunutá nižšie, správy sa neprekrývajú		

ID	7	Názov	Vloženie spra	avy medzi dve čiary života, pričom jedna už obsahuje správu		
Rozhranie Sekvenčný diagram – Vloženie správy						
Účel		Overenie	funkčnosti vytv	vorenia a zarovnania správy		
Vstupné po	dmienky		Je vytvorená	prázdna vrstva obsahujúca tri čiary života, medzi druhou a treťou	je vytvorená správa	
Výstupné p	odmienk	у	Správa je vyt neprekrývajú	vorená medzi čiarami života a je zarovnaná na najvyšiu možnú po	ozíciu, správy sa	
Krok	Akcia	1		Očakávaná reakcia	Skutočná reakcia	
1	Stlače	enie medze	rníka	Zobrazí sa kurzor		
2	Kliknu Intera	Kliknutie na tlačidlo "Add Interaction"		Zobrazia sa záložky pre všetky vrstvy, kurzor je viditeľný		
3	Výbei života vrstvy	Výber vrstvy zdrojovej čiary života kliknutím na záložku vrstvy		Všetky ostatné vrstvy sa zneviditeľnia		
4	Kliknu prvú z	Kliknutie pomocou kurzora na prvú z čiar života		Zobrazia sa záložky pre výber vrstvy cieľovej čiary života		
5	Výbei života vrstvy	Výber vrstvy cieľovej čiary života kliknutím na záložku vrstvy		Všetky ostatné vrstvy sa zneviditeľnia		
6	Kliknu druhú nižšiu existu	Kliknutie pomocou kurzora na druhú z čiar života, na pozíciu nižšiu než pozícia už existujúcej správy		Zobrazí sa dialógové okno na vyplnenie textu správy		
7	Kliknu	Kliknutie na tlačidlo "Create"		Vytvorí sa správa medzi zvolenými čiarami života a je zarovnaná na najvyššiu možnú pozíciu tak, aby neprekrývala už existujúcu správu		

ID	8	Názov	ázov Vloženie správy medzi dve čiary života a nad tretiu, ktorá je umiestnená uprostred a nachádza s nižšie než ostatné čiary života				
Rozhranie		Sekvenčn	vý diagram – V	loženie správy			
Účel		Overenie	funkčnosti vytv	vorenia a zarovnania správy a zarovnania čiary života			
Vstupné po	dmienky		Je vytvorená	prázdna vrstva obsahujúca tri čiary života (stredná je umiestnená	nižšie než ostatné)		
Výstupné p	odmienk	у	Správa je vy života je pos	tvorená medzi čiarami života a je zarovnaná na najvyšiu možnú po unutá	zíciu, stredná čiara		
Krok	Akcia	1		Očakávaná reakcia	Skutočná reakcia		
1	Stlače	enie medze	rníka	Zobrazí sa kurzor			
2	Klikni Intera	Kliknutie na tlačidlo "Add Interaction"		Zobrazia sa záložky pre všetky vrstvy, kurzor je viditeľný			
3	Výber života vrstvy	Výber vrstvy zdrojovej čiary života kliknutím na záložku vrstvy		Všetky ostatné vrstvy sa zneviditeľnia			
4	Klikni prvú č	utie pomoco čiaru života	ou kurzora na	Zobrazia sa záložky pre výber vrstvy cieľovej čiary života			
5	Výbei života vrstvy	Výber vrstvy cieľovej čiary života kliknutím na záložku vrstvy		Všetky ostatné vrstvy sa zneviditeľnia			
6	Kliknu tretiu zvole čiary najvy čiary	Kliknutie pomocou kurzora na tretiu čiaru života tak, že zvolená pozícia v rámci tejto čiary života je vyššia než najvyššia pozícia strednej čiary života		Zobrazí sa dialógové okno na vyplnenie textu správy			
7	Kliknu	utie na tlači	dlo "Create"	Vytvorí sa správa medzi zvolenými čiarami života, stredná čiara života je posunutá na nižšiu pozíciu, nová správa neprekrýva rámik strednej čiary života			

ID	9	Názov	Vloženie spra	ávy medzi dve čiary života na rôznych vrstvách	
Rozhranie	Rozhranie Sekvenčný diagram – Vloženie správy				
Účel		Overenie	funkčnosti vytv	vorenia a zarovnania správy	
Vstupné po	dmienky		Sú vytvorené	e dve vrstvy, pričom každá vrstva obsahuje jednu čiaru života	
Výstupné p	odmienk	у	Správa je vyt	tvorená medzi čiarami života a je zarovnaná na najvyšiu možnú po	ozíciu
Krok	Krok Akcia			Očakávaná reakcia	Skutočná reakcia
1	Stlače	enie medze	rníka	Zobrazí sa kurzor	
2	Kliknı Intera	utie na tlačio oction"	dlo "Add	Zobrazia sa záložky pre všetky vrstvy, kurzor je viditeľný	
3	Výber života vrstvy	r vrstvy zdro a kliknutím r /	ojovej čiary na záložku	Všetky ostatné vrstvy sa zneviditeľnia	
4	Kliknu jednu	utie pomoco z čiar život	ou kurzora na a	Zobrazia sa záložky pre výber vrstvy cieľovej čiary života	
5	Výber života vrstvy	r vrstvy cieľ a kliknutím r ⁄	ovej čiary na záložku	Všetky ostatné vrstvy sa zneviditeľnia	
6	Kliknı ďalšiu	Kliknutie pomocou kurzora na ďalšiu z čiar života		Zobrazí sa dialógové okno na vyplnenie textu správy	
7	Kliknu	utie na tlačio	dlo "Create"	Vytvorí sa správa medzi zvolenými čiarami života a je zarovnaná na najvyššiu možnú pozíciu tak, aby neprekrývala rámik žiadnej čiary života	

B.3 Fragment

ID	10	Názov	Pridanie fragmentu				
Rozhranie		Sekvenčn	ıý diagram – Fi	ragment – vytvorenie fragmetnu			
Účel		Overenie	funkčnosti vytv	vorenia nového fragmentu.			
Vstupné po	dmienky	,	Je vytvorený	diagram s čiaramy života a interakciami.			
Výstupné p	odmienk	у	Na obrazovk	e je vyditeľný fragment			
Krok	Akcia			Očakávaná reakcia	Skutočná reakcia		
1	Stlač	Stlačenie medzerníka		Zobrazí sa kurzor			
2	Klikni fragm	Kliknutie na tlačidlo "Add fragment"		Zobrazia sa záložky pre vrstvy a kurzor je viditeľný			
3	Výbe fragm	Výber vrstvy, na ktorej má byť fragment umiestnený		Všetky ostatné vrstvy sa zneviditeľnia			
4	Stlačo kde n ťahar fragm myši.	enie myšy r ná fragment nie na miest nent koniec	na mieste, t začiatok, to, kde má a pustenie	Zobrazí sa okno pre zadanie mena fragmentu.			
5	Zadanie mena fragmentu a kliknutie na tlačidlo "Create".			Vykreslenie fragmentu na obrazovke. Frament je zobrazený cez čiary života, cez ktoré sme pri ťahu prešli.			

ID	11	Názov	Názov Pridanie vnoreného fragmentu				
Rozhranie	Sekvenčný diagram – F			agment vytvorenie fragmetou			
		Oekvenich					
Učel		Overenie	funkčnosti vytv	vorenia nového fragmentu.			
Vstupné po	dmienky		Je vytvorený	diagram s čiaramy života, interakciami a fragmentom			
Výstupné p	odmienk	у	Na obrazovk	e je vyditeľný vnorený fragment			
Krok	rok Akcia			Očakávaná reakcia	Skutočná reakcia		
1	Stlač	Stlačenie medzerníka		Zobrazí sa kurzor			
2	Klikni fragm	Kliknutie na tlačidlo "Add fragment"		Zobrazia sa záložky pre vrstvy a kurzor je viditeľný			
3	Výbe fragm	Výber vrstvy, na ktorej má byť fragment umiestnený		Všetky ostatné vrstvy sa zneviditeľnia			
4	Stlačí kde n rámci miest konie a pus	Stlačenie myšy na mieste, kde má fragment začiatok, v rámci fragmentu, ťahanie na miesto, kde ma fragment koniec, v rámci fragmentu, a pustenie myši.		Zobrazí sa okno pre zadanie mena fragmentu.			
5	Zada a klik "Crea	nie mena fra nutie na tlad ite".	agmentu čidlo	Vykreslenie fragmentu na obrazovke. Frament je zobrazený cez čiary života, cez ktoré sme pri ťahu prešli.			

B.4 Mazanie Elementov

ID	12	Názov	Vymazanie s	mazanie správy				
Rozhranie		Sekvenčn	ý diagram – Vy	ymazanie				
Účel		Overenie	funkčnosti vym	nazania správy				
Vstupné po	dmienky	,	Na jednej vrs	tve je vytvorené 2 čiary života medzi ktorými je vytvorená správa				
Výstupné p	odmienk	y	Správa bola v	vymazaná				
Krok	Akcia	Akcia		Očakávaná reakcia	Skutočná reakcia			
1	Stlač	Stlačenie medzerníka		Zobrazí sa kurzor				
2	Klikn "Dele kurzo	Kliknutie na button "DeleteEment" pomocou kurzora		Zobrazia sa záložky pre všetky vrstvy, kurzor je viditeľný				
3	Výbe vytvo medz zákla zálož	Výber vrstvy na ktorej sú vytvorené 2 čiary života medzi ktorými je správa na základe kliknutia na jej záložku		Všetky vrstvy okrem vybratej sa skryjú. Vybratá zostane viditeľná a nie je vidno jej záložku				
4	Klikn	utie na sprá	vu	Správa zmizne z obrazovky a kurzor zmizne. Všetky vytvorené vrstvy sú viditeľné				

ID	13	Názov	zov Vymazanie čiary života bez správ				
Rozhranie		Sekvenčn	ý diagram – V	ymazanie			
Účel		Overenie	funkčnosti vym	nazania čiary života bez správ			
Vstupné po	dmienky	,	Na jednej vrs	Na jednej vrstve je vytvorená čiara života			
Výstupné p	odmienk	y	Čiara života	pola vymazaná			
Krok	Akcia	Akcia		Očakávaná reakcia	Skutočná reakcia		
1	Stlač	Stlačenie medzerníka		Zobrazí sa kurzor			
2	Klikn "Dele kurzo	Kliknutie na button "DeleteEment" pomocou kurzora		Zobrazia sa záložky pre všetky vrstvy, kurzor je viditeľný			
3	Výbe vytvo zákla zálož	Výber vrstvy na ktorej je vytvorená 1 čiara života na základe kliknutia na jej záložku		Všetky vrstvy okrem vybratej sa skryjú. Vybratá zostane viditeľná a nie je vidno jej záložku			
4	Klikn	utie na čiaru	ı života	Čiaru života zmizne z obrazovky a kurzor zmizne. Všetky vytvorené vrstvy sú viditeľné			

ID	14	Názov	zov Vymazanie čiary života so správami				
Rozhranie		Sekvenčn	ý diagram – V	ymazanie			
Účel		Overenie	funkčnosti vym	nazania čiary života so správami			
Vstupné podmienky		,	Existujú 2 vrstvy. Na prvej sú 2 čiary života (A,B), na druhej 1(C). Na prvej vrstve vedie z čiary života A správa do čiary života B a taktiež z B do A. Z druhej vrstvy správa z čiary života C do čiary života B vrstvy 1 a taktiež aj z B do C.				
Výstupné p	odmienk	y	Čiara života I	oola vymazaná a taktiež aj všetky správy ktoré z nej / do nej vedú			
Krok	Akcia	Akcia		Očakávaná reakcia	Skutočná reakcia		
1	Stlač	enie medze	rníka	Zobrazí sa kurzor			
2	Klikn "Dele kurzc	Kliknutie na button "DeleteEment" pomocou kurzora		Zobrazia sa záložky pre všetky vrstvy, kurzor je viditeľný			
3	Výbe vytvo zákla zálož	Výber vrstvy na ktorej sú vytvorené 2 čiary života na základe kliknutia na jej záložku		Všetky vrstvy okrem vybratej sa skryjú. Vybratá zostane viditeľná a nie je vidno jej záložku			
4	Kliknutie na čiaru života B			Čiara života zmizne z obrazovky spolu so všetkými správami s ktorými bola spojená a kurzor zmizne. Všetky vytvorené vrstvy sú viditeľné			

ID	15	Názov	Vymazanie s	Vymazanie scény				
Rozhranie	Sekvenčný diagram – Vymazanie							
Účel		Overenie	funkčnosti vym	nazania scény				
Vstupné podmienky Existujú života A správa z			Existujú 2 vrs života A asyr správa z čiar	vy. Na prvej sú 2 čiary života (A,B), na druhej 1(C). Na prvej vrstve vedie z čiary chrónna správa do čiary života B a taktiež z B do A. Z druhej vrstvy asynchrónna života C do čiary života B vrstvy 1 a taktiež aj z B do C.				
Výstupné p	odmienk	у	Všetky elementy na obrazovke sú vymazané a vznikne nová čistá vrstva					
Krok	Akcia	Akcia		Očakávaná reakcia	Skutočná reakcia			
1	Stlač	enie medze	rníka	Zobrazí sa kurzor				
2	Kliknu "Clea kurzo	Kliknutie na button "ClearScene" pomocou kurzora		Všetky elementy na obrazovke zmiznú a zostane len novo vytvorená čistá vrstva. Kurzor zmizne.				

B.5 Ukladanie a Načítavanie Schémy do/z XML

ID	16	Názov	Uloženie vrst	iev			
Rozhranie		Sekvenčn	vý diagram – V	rstva – uloženie niekoľkých vrstvy			
Účel		Overenie	funkčnosti ulož	ženia niekoľkých vrstiev			
Vstupné po	dmienky		-	-			
Výstupné p	odmienk	у	Vygenerovar	ý XML súbor obsahujúci uložený diagram, v tomto teste 4 objekty	typu vrstva		
Krok	Akcia	Akcia		Očakávaná reakcia	Skutočná reakcia		
1	Prida kliknu Layer	ridanie prvej vrstvy – liknutie na tlačidlo "Add aver"		V diagrame pribudne nová vrstva			
2	Prida v kroł	Pridanie druhej vrstvy – ako v kroku 1		V diagrame pribudne nová vrstva			
3	Prida v kroł	Pridanie tretej vrstvy – ako v kroku 1		V diagrame pribudne nová vrstva			
4	Kliknu	utie na tlačio	dlo "Save"	V adresári, kde sa nachádza aj spustiteľný program, je vytvorený .xml súbor obsahujúci export diagramu.			

ID	17	Názov	Načítanie vrs	Načítanie vrstiev				
Rozhranie Sekvenčný diagram		ý diagram – V	rstva – načítanie niekoľkých vrstiev					
Účel Overenie funkčnosti nač			funkčnosti nač	ítania niekoľkých vrstiev				
Vstupné podmienky Existujúci .>		Existujúci .xn	l súbor obsahujúci export diagramu s 4 vrstvami – výstup úspešného testu s ID 16.					
Výstupné p	odmienk	y	Vytvorený dia	igram obsahujúci 4 vrstvy – stav zodpovedá stavu pred uložením v teste s ID 16				
Krok Akcia			Očakávaná reakcia	Skutočná reakcia				
1 Kliknutie na tlačidlo "Load"		dlo "Load"	V diagrame pribudnú 4 vrstvy.					

ID	18	Názov	Uloženie vrst	Jloženie vrstvy s čiarou života			
Rozhranie		Sekvenčn	ý diagram – V	v diagram – Vrstva – uloženie vrstvy s čiarou života			
Účel		Overenie	funkčnosti ulož	ženia vrstvy s čiarou života			
Vstupné po	dmienky		-				
Výstupné podmienky		Vygenerovaný XML súbor obsahujúci uložený diagram, v tomto teste 1 vrstva s 1 čiarou života					
Krok	ok Akcia			Očakávaná reakcia	Skutočná reakcia		
1	Pridanie čiary života – kliknutie na tlačidlo "Add layer", vybratie vrstvy a vyplnenie textových políčok		vota – Ilo "Add rstvy vvých políčok	V diagrame pribudne nová čiara života so zadanými údajmi			
2 Kliknutie na tlačidlo "Save"		dlo "Save"	V adresári, kde sa nachádza aj spustiteľný program, je vytvorený .xml súbor obsahujúci export diagramu.				

ID	19	Názov	Načítanie vrstvy s čiarou života					
Rozhranie Sekvenčn			vý diagram – V	ý diagram – Vrstva – načítanie vrstvy s čiarou života				
Účel Overenie funkčnost			funkčnosti nač	ítania vrstvy s čiarou života				
Vstupné podmienky			Existujúci .xn testu s ID 18	istujúci .xml súbor obsahujúci export diagramu s 1 vrstvou a 1 čiarou života – výstup úspešného stu s ID 18.				
Výstupné podmienky		Vytvorený diagram obsahujúci 1 vrstvu a 1 čiaru života – stav zodpovedá stavu pred uložením v teste s ID 18						
Krok Akcia			Očakávaná reakcia	Skutočná reakcia				
1	1 Kliknutie na tlačidlo "Load"		dlo "Load"	V diagrame sa zobrazí vrstva, na ktorej bude čiara života s takými parametrami, aké používateľ zadal v kroku 1 testu s ID č. 18				

ID	20	Názov	Uloženie dia	gramu s interakciou		
Rozhranie		Sekvenčn	vý diagram – V	rstva – uloženie diagramu s interakciou		
Účel		Overenie	funkčnosti ulož	źenia diagramu s interakciou		
Vstupné po	dmienky		-			
Výstupné p	odmienk	у	Vygenerovar interakcia me	νý XML súbor obsahujúci uložený diagram, v tomto teste 1 vrstva, 2 čiary života a 1 edzi týmito čiarami života		
Krok Akcia			Očakávaná reakcia	Skutočná reakcia		
1	Pridanie čiary života – kliknutie na tlačidlo "Add interaction", vybratie vrstvy a vyplnenie textových políčok		vota – Ilo "Add atie vrstvy vvých políčok	V diagrame pribudne nová čiara života so zadanými údajmi		
2	Prida podot	nie druhej č one ako v k	čiary života – roku č. 1	V diagrame pribudne nová čiara života (hneď vedľa tej prvej z kroku č. 1) so zadanými údajmi		
3	3 Pridanie interakcie – kliknutie na tlačidlo "Add Interaction", následne výber výber vrstvy a počiatočnej čiary života, potom výber vrstvy a cieľovej čiary života. Nasleduje vyplnenie názvu interakcie a potvrdenie celej operácie		ie – kliknutie Interaction", výber vrstvy ny života, vy a cieľovej leduje interakcie ej operácie	Medzi čiarami života vytvorenými v krokoch č. 1 a 2 je pridaná interakcia smerujúca z čiary života z kroku č. 1 do čiary života z kroku č. 2. Interakcia má zadanňy názov.		
4 Kliknutie na tlačidlo "Save"		dlo "Save"	V adresári, kde sa nachádza aj spustiteľný program, je vytvorený .xml súbor obsahujúci export diagramu.			

ID	21	Názov	Uloženie dia	gramu s interakciou		
Rozhranie		Sekvenčn	vý diagram – V	rstva – uloženie diagramu s interakciou		
Účel		Overenie	funkčnosti ulož	źenia diagramu s interakciou		
Vstupné po	dmienky		-			
Výstupné p	odmienk	у	Vygenerovar interakcia me	ıý XML súbor obsahujúci uložený diagram, v tomto teste 1 vrstva, 2 čiary života a 1 ₂dzi týmito čiarami života		
Krok Akcia			Očakávaná reakcia	Skutočná reakcia		
1	Pridanie čiary života – kliknutie na tlačidlo "Add layer", vybratie vrstvy a vyplnenie textových políčok		vota – dlo "Add rstvy vvých políčok	V diagrame pribudne nová čiara života so zadanými údajmi		
2	2 Pridanie druhej čiary života – podobne ako v kroku č. 1		čiary života – roku č. 1	V diagrame pribudne nová čiara života (hneď vedľa tej prvej z kroku č. 1) so zadanými údajmi		
3	 Pridanie interakcie – kliknutie na tlačidlo "Add Interaction", následne výber výber vrstvy a počiatočnej čiary života, potom výber vrstvy a cieľovej čiary života. Nasleduje vyplnenie názvu interakcie a potvrdenie celej operácie 		ie – kliknutie Interaction", výber vrstvy iry života, vy a cieľovej leduje interakcie ej operácie	Medzi čiarami života vytvorenými v krokoch č. 1 a 2 je pridaná interakcia smerujúca z čiary života z kroku č. 1 do čiary života z kroku č. 2. Interakcia má zadanňy názov.		
4 Kliknutie na tlačidlo "Save"		dlo "Save"	V adresári, kde sa nachádza aj spustiteľný program, je vytvorený .xml súbor obsahujúci export diagramu.			

ID	22	Názov	Načítanie inte	Jačítanie interakcie				
Rozhranie Sekvenčn			vý diagram – V	ý diagram – Vrstva – načítanie interakcie				
Účel Overenie funkčnost				ítania interakcie				
Vstupné podmienky			Existujúci .xn medzi týmito	stujúci .xml súbor obsahujúci export diagramu s 1 vrstvou, 2 čiarami života a jednou interakciou dzi týmito čiarami života – výstup úspešného testu s ID 21				
Výstupné podmienky			Vytvorený dia stavu pred ul	[,] diagram obsahujúci 1 vrstvu, 2 čiary života a interakciu medzi nimi – stav zodpovedá J uložením v teste s ID 21				
Krok Akcia			Očakávaná reakcia	Skutočná reakcia				
1 Kliknutie na tlačidlo "Load"		dlo "Load"	V diagrame sa zobrazí vrstva, na ktorej budú čiary života a interakcia medzi nimi, s takými parametrami, aké používateľ zadal v teste s ID č. 21					

ID	23	Názov Uloženie diag		gramu s fragmentom		
Rozhranie Sekvenčný dia		vý diagram – V	γ diagram – Vrstva – uloženie diagramu s fragmentom			
Účel	Účel Overenie funkčnosti			ženia diagramu s fragmentom		
Vstupné po	dmienky		-			
Výstupné podmienky Vygenerov interakcia r			Vygenerovar interakcia me	ý XML súbor obsahujúci uložený diagram, v tomto teste 1 vrstva, 2 čiary života, 1 dzi týmito čiarami života a 1 fragment okolo interakcie		
Krok	Akcia	1		Očakávaná reakcia	Skutočná reakcia	
1	Pridanie čiary života – kliknutie na tlačidlo "Add layer", vybratie vrstvy a vyplnenie textových políčok		vota – dlo "Add rstvy ových políčok	V diagrame pribudne nová čiara života so zadanými údajmi		
2	Prida podol	Pridanie druhej čiary života – podobne ako v kroku č. 1		V diagrame pribudne nová čiara života (hneď vedľa tej prvej z kroku č. 1) so zadanými údajmi		
3	Pridanie interakcie – kliknutie na tlačidlo "Add Interaction", následne výber výber vrstvy a počiatočnej čiary života, potom výber vrstvy a cieľovej čiary života. Nasleduje vyplnenie názvu interakcie a potvrdenie celej operácie		cie – kliknutie Interaction", výber vrstvy ny života, tvy a cieľovej leduje interakcie ej operácie	Medzi čiarami života vytvorenými v krokoch č. 1 a 2 je pridaná interakcia smerujúca z čiary života z kroku č. 1 do čiary života z kroku č. 2. Interakcia má zadanňy názov.		
4 Pridanie fragmentu – kliknutie na tlačidlo "Add Fragment" a vybratie dvoch bodov na vrstve – ľavý horný roh fragmentu a pravý dolný.		ntu – kliknutie Fragment" bodov na ný roh ⁄ý dolný.	Vykreslený diagram, ktorý "obaľuje" čiary života a interakciu medzi nimi, ktoré boli pridané v krokoch č. 1, 2 a 3.			
5	5 Kliknutie na tlačidlo "Save"			V adresári, kde sa nachádza aj spustiteľný program, je vytvorený .xml súbor obsahujúci export diagramu.		

ID	24	Názov	Načítanie fra	lačítanie fragmentu				
Rozhranie		Sekvenčn	ý diagram – V	m – Vrstva – načítanie fragmentu				
Účel		Overenie	funkčnosti nač	ítania fragmentu				
Vstupné podmienky Existujúci medzi týr testu s ID			Existujúci .xn medzi týmito testu s ID 23	nl súbor obsahujúci export diagramu s 1 vrstvou, 2 čiarami života, jednou interakciou o čiarami života a fragment obaľujúci predchádzajúce objekty – výstup úspešného s				
Výstupné podmienky			Vytvorený dia stav zodpove	ný diagram obsahujúci 1 vrstvu, 2 čiary života, interakciu medzi nimi a obaľujúci fragment – dpovedá stavu pred uložením v teste s ID 23				
Krok Akcia			Očakávaná reakcia	Skutočná reakcia				
1	Kliknutie na tlačidlo "Load"		dlo "Load"	V diagrame sa zobrazí vrstvy, na ktorej bude čiara života, interakcia medzi nimi a obaľujúci fragment, s takými parametrami, aké používateľ zadal v teste s ID č. 23				

C Report z Testovania

Testovacia fáza projektu prebiehala počas jedného šprintu. Boli vybrané viaceré typy používateľov, ktorý vykonávali pripravené scenáre použitia.

Prvou skupinou boli používatelia, pre ktorých je koncept našej aplikácie primárne určený. Znalci UML, ktorý mali ukázať, či im aplikácia vyhovuje používateľsky a funkčne. Tejto skupine sme priložili najväčšiu váhu a pokladáme ju za najrelevantnejšiu.

Typický používateľ druhej skupiny je študent, iba s miernymi skúsenosťami UML. V tejto skupine sme sa zamerali skôr na používateľskú stránku. Ako sa s aplikáciou pracuje. Čo by malo byt intuitívnejšie. Či je všetko chápané tak ako sme si mysleli. Takýto prístup nám umožnil nenahraditeľnú spätnú väzbu od nezainteresovaného používateľa napriek tomu, že nie je cieľovým používateľom.

Používatelia vykonávali predpripravené testy pod dozorom člena tímu, ktorý zaznamenával informácie nad rámec hraníc testov. Príkladom môžu byť slovne komentáre na možné rozšírenia, ktoré by sa používateľom páčili.

testov	24
počet iterácii	3
úspešné testy	20
neúspešné testy	4

Z výsledkov testov vyplynuli niektoré drobné chyby, ktoré boli takmer okamžite opravené a nasadené. Kritickejšie chyby nastali pri testovaní načítavania. Ani jeden test načítania neprebehol správne. Táto chyba je spojená so zapracovávaním metamodelu, ktoré znefunkčnilo túto funkcionalitu. Ostatné testy prebehli úspešne.

Z použiteľných návrhov na zmenu vyplynulo označovanie záložiek podobne ako je tomu v prípade označovania čiary života. Používateľom sa stávalo, že sa nevedeli trafiť do správnej záložky ako boli pohľadom mimo správneho uhla.

D Preberací protokol

PREBERACÍ PROTOKOL ZA ZIMNÝ SEMESTER

Tímový projekt 2013/2014

Tím 02 – GAMATEPI

Predmet odovzdávania:

- Dokumentácia k riadeniu projektu verzia z letného semestra
- Dokumentáciu k inžinierskemu dielu verzia z letného semestra

Vedúci projektu: Ing. Ivan Polášek, PhD Podpisom potvrdzuje prebratie vyššie uvedených častí projektu.

V Bratislave

.....

.....

Dátum

Podpis