



**ČESKÉ VYSOKÉ
UČENÍ TECHNICKÉ
V PRAZE**

F3

**Fakulta elektrotechnická
Katedra počítačové grafiky a interakce**

Bakalářská práce

Uživatelský průzkum a UX recenze aplikace I3T. Návrh a vytvoření prototypů pro vylepšení použitelnosti aplikace I3T.

Barbora Hálová

Květen 2025

Vedoucí práce: Ing. Petr Felkel, Ph.D.

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Hálová** Jméno: **Barbora** Osobní číslo: **508502**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra počítačové grafiky a interakce**
Studijní program: **Otevřená informatika**
Specializace: **Počítačové hry a grafika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Uživatelský průzkum a UX recenze aplikace I3T. Návrh a vytvoření prototypů pro vylepšení použitelnosti aplikace I3T.

Název bakalářské práce anglicky:

User research and UX review of the I3T app. Design and prototyping to improve the usability of the I3T app.

Pokyny pro vypracování:

Vytvořte si výzkumný plán a poté proveďte kvalitativní průzkum uživatelů nástroje na Interaktivní výuku transformací (I3T). Proveďte též rešerši uživatelských rozhraní podobných programů. Na základě sesbíraných dat a rešerše předchozích prací studentů proveďte expertní analýzu uživatelské přívětivosti stávajícího designu a navrhnete změny uživatelského rozhraní dle výsledků. Všechny navržené změny otestujte pomocí prototypů s uživateli a vytvořte výsledný návrh k implementaci.

Seznam doporučené literatury:

- [1] Steve Krug. Web design : nenuťte uživatele přemýšlet! Computer Press, 2006.
- [2] Elizabeth Goodman, Mike Kuniavsky a Andrea Moed. Observing the User Experience, Second Edition: A Practitioner's Guide to User Research. 2nd vyd. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2012. ISBN 0123848695.
- [3] Vít Zadina. „Testování užitečnosti nástroje pro výuku transformací“. <http://hdl.handle.net/10467/83400>. Bakalářská práce. FIT ČVUT, 2020

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Petr Felkel, Ph.D. Katedra počítačové grafiky a interakce

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: _____ Termín odevzdání bakalářské práce: _____

Platnost zadání bakalářské práce: **20.09.2026**

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Studentka bere na vědomí, že je povinna vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studentky

PROHLÁŠENÍ

Já, níže podepsaná

Příjmení, jméno studenta: Hálová Barbora
Osobní číslo: 508502
Název programu: Otevřená informatika

prohlašuji, že jsem bakalářskou práci s názvem

Uživatelský průzkum a UX recenze aplikace I3T. Návrh a vytvoření prototypů pro vylepšení použitelnosti aplikace I3T.

vypracovala samostatně a uvedla veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací a Rámcovými pravidly používání umělé inteligence na ČVUT pro studijní a pedagogické účely v Bc a NM studiu.

Prohlašuji, že jsem v průběhu příprav a psaní závěrečné práce použila nástroje umělé inteligence. Vygenerovaný obsah jsem ověřila. Stvrzuji, že jsem si vědoma, že za obsah závěrečné práce plně zodpovídám.

V Praze dne 22.05.2025

Barbora Hálová

.....
podpis studenta

Poděkování / Prohlášení

Chtěla bych poděkovat své rodině a přátelům za podporu během studia a Ing. Petru Felkelovi, Ph.D. za cenné rady a pomoc při tvorbě práce. Také děkuji týmu, který se mnou podílel na vývoji aplikace I3T a všem ochotným studentům, kteří mi pomohli s testováním.

Prohlašuji, že jsem bakalářskou práci vypracovala samostatně a uvedla veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací a Rámcovými pravidly používání umělé inteligence na ČVUT pro studijní a pedagogické účely v Bc a NM studiu. Prohlašuji, že jsem v průběhu příprav a psaní závěrečné práce použila nástroje umělé inteligence. Vygenerovaný obsah jsem ověřila. Stvrzuji, že jsem si vědoma, že za obsah závěrečné práce plně zodpovídám.

Abstrakt / Abstract

Práce se zabývá vylepšením použitelnosti programu Interactive Tool for Teaching Transformations (I3T), který slouží jako výuková pomůcka při vysvětlování geometrických transformací v počítačové grafice. Cílem práce bylo navrhnout a následně ověřit úpravy uživatelského rozhraní, které zvýší intuitivnost ovládání, sníží kognitivní zátěž studentů a zajistí pozitivní uživatelskou zkušenost, čímž bude podpořena motivace studentů k využívání nástroje během výuky.

Na základě rešerše v oblasti použitelnosti a uživatelského designu byla zvolena kvalitativní metoda testování použitelnosti s uživateli a princip iterativního návrhu. V úvodní fázi bylo provedeno testování s cílem identifikovat klíčové problémy v interakci uživatelů s aplikací. Na základě zjištěných nedostatků byly navrženy konkrétní úpravy, které byly konzultovány a dále rozvíjeny ve spolupráci s vývojovým týmem I3T. Následně byl zahájen cyklus iterativního testování, v jehož průběhu byly navržené úpravy ověřovány a dále rozvíjeny s cílem postupně odstraňovat dříve identifikované nedostatky, reagovat na nově nalezené problémy a celkově zlepšovat uživatelské rozhraní aplikace.

Výsledkem práce je přehlednější a efektivněji ovladatelná aplikace, která na základě zvolené metriky dosáhla zlepšení ve vybraných kategoriích o 30–60%. Součástí práce jsou také doporučení pro další rozvoj aplikace se zaměřením na zvyšování uživatelské přívětivosti.

Klíčová slova: I3T, počítačová grafika, testování použitelnosti, použitelnost, iterativní design

The Thesis focuses on improving the usability of the Interactive Tool for Teaching Transformations (I3T), an educational tool designed to support the explanation of geometric transformations in computer graphics. The aim of the thesis was to propose and subsequently verify user interface modifications that enhance intuitive control, reduce students' cognitive load, and provide a positive user experience, thereby encouraging students to use the tool during their learning process.

Based on research in the field of usability and user-centered design, a qualitative usability testing method with users and the principle of iterative design were chosen. In the initial phase, usability testing was conducted to identify key issues in user interaction with the application. Based on the identified challenges, specific improvements were proposed, which were further developed in collaboration with the I3T development team. This was followed by a cycle of iterative testing, during which the proposed modifications were verified and further refined to gradually eliminate previously identified issues, respond to newly discovered problems, and continuously improve the application's user interface.

The outcome of this thesis is a more user-friendly and efficiently operable application which achieved improvements of 30–60% in selected categories based on the chosen metric. The thesis also includes recommendations for future development in the area of usability.

Keywords: I3T, computer graphics, usability testing, usability, iterative design

Obsah /

1 Úvod	1	
1.1 Motivace ke zlepšení aplikace . . .	1	
1.2 Popis workspace	2	
1.2.1 Uzly	2	
1.2.2 Sekvence	3	
1.2.3 Transformace	4	
1.2.4 Operátory	4	
1.2.5 Kamera	4	
1.2.6 Obrazovka	4	
1.2.7 Model	5	
1.2.8 Cyklus	5	
1.2.9 Puls	5	
2 Použitelnost a uživatelské testování v literatuře	6	
2.1 Použitelnost	6	
2.1.1 Proč je použitelnost důležitá	6	
2.1.2 Jak použitelnost zlepšit	6	
2.2 Iterativní design	7	
2.3 Paralelní design	7	
2.4 Testování uživatelského rozhraní	8	
2.4.1 Vývoj uživatelského testování	8	
2.4.2 Co jsou testy použitelnosti	9	
2.4.3 Co a kdy testovat	10	
2.4.4 Jak vybrat účastníky testu	10	
2.4.5 Jak testování připravit	10	
2.4.6 Průběh testování	11	
2.4.7 Zpracování výsledků	11	
3 Úvodní testování použitelnosti I3T	12	
3.1 Předchozí testování	12	
3.2 Úvodní testování	13	
3.2.1 Metodologie	14	
3.2.2 Výsledky	14	
4 Nalezené problémy	16	
4.1 Celý pracovní prostor	16	
4.2 Menu	16	
4.3 Dráty	16	
4.4 Uzly	17	
4.5 Úhly	18	
4.6 Sekvence	19	
4.7 Krokování (<i>tracking</i>)	19	
4.8 Kamera	19	
4.9 Model	20	
4.10 Cyklus	20	
4.11 Zpracování chyb (<i>error handling</i>)	20	
5 Návrh změn	21	
5.1 Konzultace s týmem	21	
5.2 Rešerše uživatelských rozhraní podobných programů	22	
5.3 Vytvořené návrhy	25	
5.3.1 Celý pracovní prostor	25	
5.3.2 Menu	26	
5.3.3 Dráty	28	
5.3.4 Uzly	28	
5.3.5 Úhly	30	
5.3.6 Sekvence	31	
5.3.7 Krokování (<i>tracking</i>)	31	
5.3.8 Kamera	31	
5.3.9 Model	32	
5.3.10 Cyklus	32	
5.3.11 Zpracování chyb (<i>error handling</i>)	33	
5.3.12 Operátory	33	
5.3.13 Další návrhy	33	
6 Iterativní testování	34	
6.1 První etapa testování	34	
6.1.1 Co bylo testováno	34	
6.1.2 Prototyp	35	
6.1.3 Průběh testování	35	
6.1.4 Výsledky	36	
6.2 Druhá etapa testování	37	
6.2.1 Nové návrhy	37	
6.2.2 Co bylo testováno	38	
6.2.3 Průběh testování	38	
6.2.4 Výsledky	39	
6.2.5 Porovnání výsledků s úvodním testováním	40	
6.3 Třetí etapa testování	41	
6.3.1 Co bylo testováno	41	
6.3.2 Průběh testování	42	
6.3.3 Výsledky	43	
7 Přehled změn v programu	45	
8 Diskuze	49	
8.1 Návrhy do budoucna	50	
8.1.1 Návrhy k implementaci a dalšímu testování	50	
8.1.2 Oblasti ke zlepšení	50	

9 Závěr	52
Literatura	53
A Scénáře testování	55
A.1 Úvodní testování 18.–22.11. . .	55
A.2 Testování 15.-16.4.	56
A.3 Testování 7.5	56
B Obsah elektronické přílohy	58

Tabulky / Obrázky

2.1	Tabulka porovnávající tradiční a zjednodušené testování. . . .9	1.1	Workspace.2
6.1	Tabulka porovnávající výsledky úvodního testování a druhého kola iterativního testování. 41	1.2	Přidávací menu.3
		1.3	Režimy zobrazení translace.3
		1.4	Operátor matice a operátor pro výpočet determinantu.4
		1.5	Cyklus5
		2.1	Graf zobrazující poměr počtu uživatelů a nalezených chyb.....8
		3.1	Vzhled původního rozhraní ve Foltově programu..... 12
		4.1	Vykreslení drátu přes uzly. 17
		4.2	Kontextové menu modelu. 17
		4.3	Menu pro výběr často používaných hodnot. 18
		4.4	Zapojení uzlu obrazovky do kamery a implicitní zapojení kamery. 19
		4.5	Chybová hláška při neshodě typů napojení..... 20
		4.6	Chybné hodnoty uvnitř uzlu lookAt transformace..... 20
		5.1	Vzhled pracovní plochy v programu Blender. 22
		5.2	Vzhled ikonového menu v programu Blender. 23
		5.3	Vzhled pracovní plochy v programu Maya..... 24
		5.4	Vzhled uzlů nástroje Blueprint v Unreal Engine..... 24
		5.5	Vzhled uzlů v editoru materiálů v Unreal Engine 25
		5.6	Návrh ikonového menu. 26
		5.7	Seřazení menu z testování..... 27
		5.8	Výsledný návrh přidávacího menu. 28
		5.9	Návrh vzhledu uzlu translace. . 29
		5.10	Návrh vzhledu napojení a sbaleného režimu. 30
		5.11	Návrh rozhraní pro zadávání úhlu. 30
		5.12	Návrh sekvence. 31
		5.13	Návrh ikon kamery pro uzly. .. 32
		5.14	Návrh validace hodnot v matici..... 33
		6.1	Papírový prototyp I3T 35
		6.2	Možnosti vzhledu napojení 37

6.3	Možnosti vzhledu tlačítka pro přepínání režimů.	38
6.4	Nový systém krokování.	41
6.5	Krokování kamery.	42
6.6	Testované ikony.	44
7.1	Finální vzhled Přidávacího menu.	45
7.2	Finální vzhled uzlu.	46
7.3	Nová nápověda pro špatné zapojení uzlů.	47
7.4	Nový vzhled prázdné sekven- ce.	47
7.5	Finální vzhled uzlů kamery a obrazovky.	48
A.1	Scéna k úkolu 3.	56
A.2	Scéna k úkolu 1.	57
A.3	Scéna k úkolu 2.	57

Kapitola 1

Úvod

I3T (Interactive Tool for Teaching Transformations) je program, který slouží k vizualizaci 3D transformací nejen v oblasti počítačové grafiky, ale i v dalších technických oborech. V současné době je využíván především v předmětu Programování grafiky, kde slouží jako doplňkový výukový nástroj pro vizualizace konceptů programování počítačové grafiky probíraných na přednáškách. Program je rovněž volně dostupný studentům pro samostatné procvičování probírané látky.

Dle mého názoru má I3T velký potenciál nejen pro usnadnění pochopení transformací v počítačové grafice, ale i v několika dalších oborech. Ve své práci se zaměřuji na vylepšení uživatelského rozhraní a ovládání aplikace tak, aby práce s ní byla efektivní, co nejjednodušší a tento potenciál mohl být plně využit.

Konkrétně se zaměřuji na část programu označovanou jako *workspace*. Jedná se o jednu z hlavních částí aplikace, kde uživatelé pracují s uzly, které mezi sebou spojují a manipulují s nimi, čímž vytváří graf scény. Mým cílem je tuto část aplikace zpřehlednit a usnadnit její používání tak, aby vyhovovala potřebám jak začínajících, tak pokročilých uživatelů.

Ve práci jsem se zaměřila především na cílovou skupinu studentů, kteří se seznamují se základy počítačové grafiky a využívají daný nástroj jako pomůcku pro lepší pochopení probíraných konceptů. Vedle studentů jsem se také okrajově zaměřila i na skupinu učitelů.

1.1 Motivace ke zlepšení aplikace

První nápad na vytvoření programu pro výuku transformací vznikl v roce 2015 jako semestrální projekt a později diplomová práce Michala Foltý [1]. Cílem bylo vytvořit jednotný nástroj, který bude sloužit jako pomůcka při výuce a jako podpora pro samostudium studentů. Program se setkal s velkým úspěchem jak u studentů tak u vyučujících. Ale měl několik nedostatků. Hlavním z nich byla přílišná provázanost uživatelského rozhraní každého uzlu s vnitřní logikou aplikace.

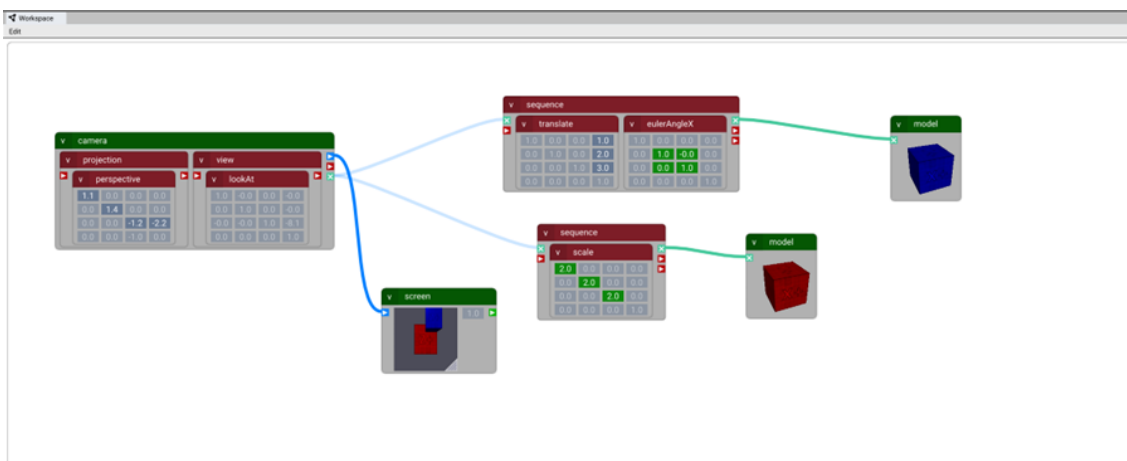
Jako reakce na nedostatky původní verze započala práce na nové verzi. Bylo navrženo nové jádro aplikace, které bylo odděleno od uživatelského rozhraní. Vznikla 3D scéna s manipulátory a sada tutoriálů seznamující uživatele s funkcemi a ovládáním programu. To, že jednotlivé části nové verze aplikace vznikaly jako součást několika bakalářských a diplomových prací, vedlo k tomu, že mnohé funkce byly přidávány spíše v pořadí jejich vývoje, než s ohledem na uživatelskou přívětivost. A použitelnosti se zatím nikdo do hloubky nevěnoval.

Užitečností nástroje se ve své Bakalářské práci zabýval Jiří Boček, který zjistil, že použití nástroje ve výuce má pozitivní dopad na pochopení transformací [2]. Tento výsledek dále zdůrazňuje potřebu zlepšit použitelnost nástroje, aby mohl být jeho vzdělávací potenciál plně využit. Vylepšení použitelnosti by také mělo zajistit, aby noví uživatelé, zejména studenti, nebyli hned na začátku demotivováni neintuitivním ovládáním či nepřehledným rozhraním.

I přesto, že aplikace obsahuje tutoriály, které mimo jiné seznamují uživatele s fungováním programu (První z těchto tutoriálů seznamuje uživatele právě s hlavními částmi pracovní plochy a jejich ovládáním), považuji za důležité věnovat se zlepšení použitelnosti samotného programu tak, aby uživatelé nebyli nuceni procházet celé tutoriály při hledání konkrétní informace, nebo při návratu k aplikaci po delší době. Cílem je vytvořit intuitivní uživatelské rozhraní, které minimalizuje pravděpodobnost chyb a umožňuje snadné a efektivní ovládání.

1.2 Popis workspace

I3T využívá systém uzlů, kde každý uzel reprezentuje část sekvence transformací aplikovaných na model při jeho zobrazování na obrazovku, nebo model samotný. Kromě toho obsahuje uzly určené pro operace s maticemi, vektory a kvaterniony, stejně jako uzly pro cykly, které umožňují rozpořehovat scénu. Vkládání objektů a manipulaci s nimi zajišťuje pracovní plocha (*workspace*) viditelná na obrázku 1.1, která je ve výchozím nastavení umístěna ve spodní části obrazovky. Uživatel si však může rozložení aplikace přizpůsobit a přesunout okno na libovolné místo podle svých potřeb. V následující sekci popisují verzi programu v2.0.0-rc3 z listopadu 2024, což je stav před začátkem práce na změnách.

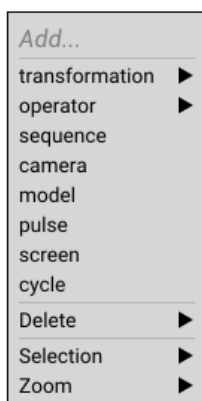


Obrázek 1.1. Workspace ve verzi v2.0.0-rc3.

Workspace je reprezentován velkou prázdnou plochou určenou k práci s uzly. V pravém horním rohu se nachází menu „Edit“, které nabízí funkci pro výběr všech uzlů a funkci pro jejich zaměření. Pro přidání nového uzlu musí uživatel kliknout pravým tlačítkem myši do pracovní plochy, čímž se zobrazí přidávací menu (obrázek 1.2). Z tohoto menu lze následně vybrat a vložit požadovaný uzel.

1.2.1 Uzly

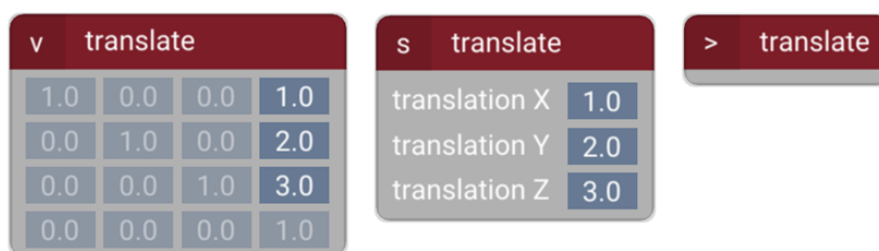
Uzly v I3T jsou rozděleny do několika typů podle jejich použití. Červené uzly reprezentují transformace, modré uzly slouží jako operátory a zelené uzly zahrnují ostatní funkce. Většina uzlů, s výjimkou transformací, obsahuje také napojení - piny. A to pro vstup (*input*) na levé straně a pro výstup (*output*) na pravé straně. Tyto piny se liší typem objektu (například matice nebo vektor), který je možné napojit a barvou, která je mu přidělena. Speciálním případem pinu je maticové násobení, které umožňuje skládat transformace a aplikovat je na modely. Některé uzly, jako například uzly pro sekvenční



Obrázek 1.2. Přidávací menu ve verzi v2.0.0-rc3.

nebo kameru, navíc umožňují vkládání dalších uzlů, konkrétně uzlů transformací. Při vložení více uzlů je na ně automaticky aplikováno maticové násobení.

Každý uzel lze zobrazit v několika režimech (obrázek 1.3), které lze měnit kliknutím na ikonu v levém horním rohu uzlu. Hlavní režimy zahrnují plné zobrazení a skryté zobrazení. Některé uzly nabízejí také zjednodušený režim, který slouží k zadání definujících parametrů matice, například u translačních matic je možné zobrazit pouze posuvný vektor místo celé matice.



Obrázek 1.3. Režimy zobrazení translace.

Každý uzel obsahuje své vlastní kontextové menu, které se zobrazí pravým kliknutím na horní část uzlu. Toto menu umožňuje uzly přejmenovat, duplikovat, smazat a provádět další akce specifické pro daný typ uzlu (popsáno níže).

1.2.2 Sekvence

Sekvence *sequence* je uzel, který umožňuje zprostředkovat transformace tím, že do něj lze vkládat transformační matice. Sekvence lze za sebe napojit prostřednictvím maticového násobení. Kromě toho lze do sekvence připojit matici jako vstup, což způsobí, že je její hodnota do sekvence zkopírována. Sekvenci lze také použít jako vstup pro jinou sekvenci nebo matici. Jsou dvě možnosti, jak to lze udělat. Pin umístěný nahoře zkopíruje jen hodnotu samotné sekvence a pin umístěný dole zkopíruje hodnotu všech napojených transformací směrem doleva až do kořene.

Každá sekvence obsahuje funkci krokování (*tracking*), která umožňuje sledovat postupné aplikování transformací buď zleva, nebo zprava. Aktuální stav objektu se při krokování zobrazuje ve scéně *scene view*. Funkci krokování lze aktivovat z menu uzlu

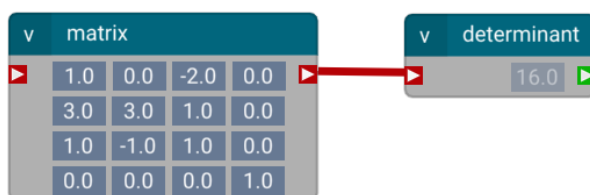
a nastavit zda bude krokování plynulé nebo ne. V tomto menu lze také nastavit počet desetinných míst pro zobrazené hodnoty a přepínat mezi dostupnými režimy zobrazení.

1.2.3 Transformace

Aplikace nabízí celkem šest typů transformací: volnou (*free*) transformaci, translaci, rotaci, škálování, lookAt a projekci. Rotace se dále dělí na rotaci kolem jednotlivých souřadnicových os, rotaci kolem konkrétní osy otáčení a rotaci pomocí kvaternionů. Transformace projekce se dělí na ortografickou, perspektivní a frustum. Uzel transformace nemá žádný vstup ani výstup, a proto musí být vložen do uzlu sekvence nebo kamery, aby mohl být použit. Každá transformace má vlastní menu, které umožňuje nastavit počet desetinných míst, režim zobrazení, umožňuje zamknout hodnoty, uložit aktuální hodnotu matice a později stávající hodnotu resetovat na tu uloženou. Některé transformace také umožňují nastavení synergie, například pro uniformní škálování a rotační matice kolem os.

1.2.4 Operátory

Operátory v aplikaci slouží k provádění různých výpočtů, jako je inverze, převody mezi rozdílnými typy objektů a další matematické operace, například výpočet determinantu zobrazený na obrázku 1.4. Na rozdíl od transformací nelze operátory vkládat do sekvencí. Podobně jako u transformací je i u operátorů možné nastavit počet zobrazovaných desetinných míst.



Obrázek 1.4. Operátor matice a operátor pro výpočet determinantu.

1.2.5 Kamera

Kamera je rozdělena na dvě hlavní části: projekční matici (*projection*) a pohledovou matici (*view*). Do každé z těchto sekcí lze vložit jednu nebo více matic, které reprezentují dané transformace. Pokud není do některé sekce vložena žádná matice, použije se jako výchozí, matice jednotková. Uzel kamery lze napojit na uzel obrazovky *screen*. Kromě toho se u kamery vizualizuje pomyslné propojení na všechny kořeny transformací. Tím se ilustruje, jak vypadá sekvence transformací při zobrazování objektu na obrazovku. U kamery je možné nastavit, jak se bude zobrazovat v *scene view*, konkrétně co bude viditelné, jakou barvu bude mít frustum a zda se bude zobrazovat jen jako obrys nebo i s barevnou výplní.

1.2.6 Obrazovka

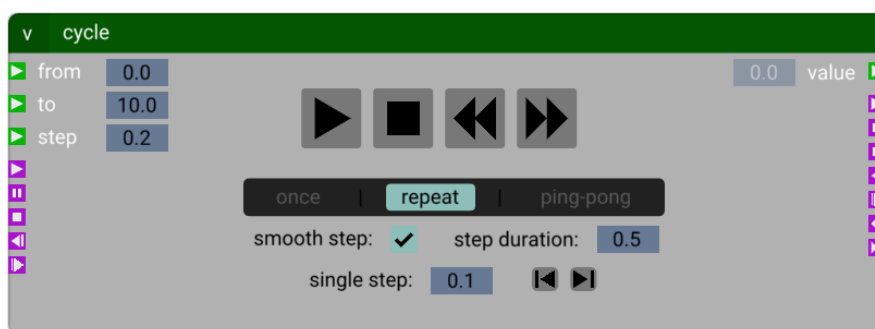
Uzel obrazovky (*screen*) slouží k zobrazení toho, co vidí kamera. Zároveň vizualizuje poměr šířky a výšky zobrazení (*aspect ratio*), který je výstupem uzlu. Aspekt se nastavuje pohybem levého dolního rohu. Vstupem uzlu je pin napojení na kameru.

1.2.7 Model

Model slouží k vizualizaci transformací, které na něj lze aplikovat prostřednictvím pinu maticového násobení. Model se zobrazuje v *scene view* na pozici, která odpovídá výsledku aplikovaných transformací. V menu lze upravit barvu a průhlednost modelu. Také je možné změnit vzhled modelu, dostupnými modely jsou: šipky, osy, králík, kachnička, konvička, koule a plocha. I3T navíc nabízí uživateli možnost nahrát vlastní model.

1.2.8 Cyklus

Cyklus (obrázek 1.5) umožňuje rozpořádání scény generováním posloupnosti číselných hodnot na základě zadaných parametrů. Je možné nastavit spodní a horní hranici hodnot a délku kroku. Cyklus postupně generuje hodnoty až do dosažení horní hranice. Uživatel si může zvolit režim generování a to buď jednorázový, opakovaný, nebo tam a zpět. Také je možné nastavit, jestli budou jednotlivé kroky diskrétní nebo spojitě (přesněji, tak malé, že působí spojitě). Vygenerovaná hodnota je dostupná prostřednictvím horního výstupu cyklu. Zbývající vstupy a výstupy cyklu jsou typu puls, který slouží k řízení průběhu cyklu, například umožňuje sériově zapojit více cyklů tak, aby se druhý spustil až po dokončení prvního.



Obrázek 1.5. Cyklus.

1.2.9 Puls

Puls slouží k ovládání cyklu. Připojením pulsu k cyklu lze jedním kliknutím spustit, zastavit, restartovat nebo zrychlit běh cyklu v závislosti na tom, kam je puls zapojen. Ve své práci jsem se cyklu, pulsu a skriptovacímu uzlu věnovala pouze okrajově. Zaměřila jsem se spíše na části programu, které budou studenti nejčastěji využívat.

Kapitola 2

Použitelnost a uživatelské testování v literatuře

Tato kapitola se věnuje představení pojmů a postupů v oblasti použitelnosti a testování uživatelských rozhraní. Popisuje důležité pojmy, jako je použitelnost, iterativní a paralelní přístup k návrhu rozhraní a podrobně rozebírá metodu testování použitelnosti. Poskytuje přehled osvědčených postupů, které lze využít při návrhu a vývoji intuitivních a efektivních uživatelských rozhraní, a na něž jsem ve své práci navazovala.

2.1 Použitelnost

Použitelnost vyjadřuje, jak snadné je uživatelské rozhraní ovládat. Je definována pěti klíčovými aspekty:

Naučitelnost (*Learnability*) udává, jak snadno se noví uživatelé dokážou naučit základní operace a začít s rozhraním efektivně zacházet.

Efektivita (*Efficiency*) popisuje, jak rychle jsou uživatelé schopni plnit úkoly poté, co si osvojili práci s rozhraním.

Zapamatovatelnost (*Memorability*) určuje, jak dobře si uživatelé dokážou po delší pauze vybavit způsob používání rozhraní a navázat na předchozí zkušenosti a dovednosti.

Chybovost (*Errors*) zahrnuje počet a závažnost chyb, kterých se uživatelé dopouští, a také schopnost se z těchto chyb zotavit.

Spokojenost uživatele (*Satisfaction*) reflektuje subjektivní hodnocení ze strany uživatelů, zejména jak pohodlné a příjemné je používání rozhraní. Také může zahrnovat, jak se uživatelům líbí vizuální stránka rozhraní. [3]

2.1.1 Proč je použitelnost důležitá

Pokud uživatelské rozhraní není dostatečně použitelné, uživatelé ho pravděpodobně přestanou využívat a začnou hledat alternativy, které lépe odpovídají jejich potřebám. U webových stránek, které nabízejí nějaký produkt, to může vést ke ztrátě zákazníků a tím ke ztrátě příjmů. V případě interních systémů se nedostatečná použitelnost negativně projeví na produktivitě zaměstnanců a celkové efektivitě pracoviště. V případě výukových systémů může nízká úroveň použitelnosti místo k pochopení probírané látky vést spíše ke zmatení uživatele, což může přispět ke ztrátě zájmu o danou problematiku. [3]

2.1.2 Jak použitelnost zlepšit

Pro získání přehledu o použitelnosti rozhraní je vhodné využít testování s reálnými uživateli. Které se skládá ze tří hlavních kroků: Prvním krokem je výběr účastníků testování, kteří odpovídají cílové skupině daného produktu. Následně je třeba připravit sadu úkolů, jež odpovídají typickému používání rozhraní a sledovat, jak uživatelé

s rozhraním interagují, co se jim daří a s čím naopak mají problémy. Klíčové je, aby uživatelé řešili úkoly samostatně. Jakékoliv zásahy ze strany pozorovatele mohou vést ke zkreslení výsledků a ztrátě cenných informací. Ověřování použitelnosti by zároveň nemělo být odkládáno až na finální fázi vývoje, kdy už je produkt plně implementován. V tomto případě mohou být potřebné změny značně náročné na opravu a zasahovat do samotné architektury systému.

Použitelnost je důležitým faktorem ve všech fázích návrhového procesu. Ještě před zahájením práce na novém návrhu je vhodné analyzovat původní řešení (pokud nějaké existuje) nebo konkurenční produkty s cílem odhalit časté problémy a identifikovat části, kterým je potřeba se věnovat. Dalším užitečným krokem je studie v terénu (*field study*), během níž je sledováno chování uživatelů při používání systému v jejich přirozeném prostředí. Následuje testování návrhů a prototypů, přičemž výsledky těchto testů slouží jako podklad pro jejich další úpravy a zlepšení. Tento cyklický postup, založený na opakovaném testování a ladění návrhu, se nazývá iterativní design. [3]

2.2 Iterativní design

Ani zkušený tým odborníků na použitelnost nedokáže navrhnout zcela bezchybné uživatelské rozhraní hned napoprvé. Právě proto by měl být proces návrhu založen na principu iterace. Iterativní design představuje postup, při němž dochází k postupným úpravám návrhu na základě zpětné vazby získané prostřednictvím uživatelského testování a dalších evaluačních metod. Po každém kole testování jsou identifikovány problémy, které uživatelé zaznamenali, a ty jsou řešeny v další verzi rozhraní. Tento přístup je považován za nejjednodušší, nejefektivnější a nákladově nejvýhodnější způsob, jak systematicky zlepšovat návrh produktu. Zároveň se jedná o jednu z nejstarších a nejdůležitějších metod v rámci přístupu User-centered design (UCD), tedy návrhu orientovaného na uživatele. [4]

Optimálně by měly být realizovány alespoň dvě iterace návrhu, což odpovídá třem verzím produktu - původnímu návrhu a dvěma následným úpravám. Pokud časové a finanční možnosti dovolují testovat každý týden, je vhodné provést i 5 až 10 iterací. Pro zajištění maximální efektivity v omezeném časovém rámci je vhodné v počátečních fázích využívat papírové nebo jiné nízkourovňové (*low fidelity*) prototypy, které umožňují rychlé testování a snadné úpravy. Teprve po odstranění nejvýraznějších problémů, které často brání odhalení detailnějších nedostatků, je vhodné přejít k podrobnějším prototypům. Tím se minimalizuje riziko nákladných změn v pozdějších fázích vývoje.

Nevýhodou iterativního designu může být přílišné zaměření na jedno konkrétní řešení. V průběhu iterací se návrh postupně vylepšuje, aby byl co nejlépe použitelný, avšak hrozí, že budou přehlédnuta jiná vhodnější řešení. Tento problém může pomoci překonat paralelní design. Tento přístup umožňuje srovnání různých cest a v některých případech vede k nalezení lepšího řešení. [5]

2.3 Paralelní design

Paralelní design je přístup, při kterém se vytváří více alternativních návrhů řešících stejný problém, které jsou následně mezi sebou porovnávány prostřednictvím uživatelského testování. Může se jednat o kompletní návrh celého rozhraní, nebo pouze o konkrétní komponentu či funkci. Optimální počet variant se pohybuje kolem tří, přičemž by nemělo být testováno více než pět verzí najednou, aby nedocházelo k zahlcení testujících a ke zkreslení výsledků. Každému účastníkovi testování by měly být během testování

předloženy dvě až tři různé varianty. Je přitom důležité měnit pořadí, ve kterém jsou návrhy prezentovány, protože první dojem může ovlivnit vnímání i následujících variant. Cílem testování přitom nemusí být pouze výběr „vítězná“ verze, ale i nalezení silných stránek jednotlivých návrhů, které lze následně zkombinovat do nového, vylepšeného řešení. Na paralelní design pak navazuje iterativní design, který umožňuje vybraný návrh dále ladit a optimalizovat na základě další zpětné vazby. [5]

2.4 Testování uživatelského rozhraní

Pokud vývojář, designér nebo někdo další, kdo se na produktu podílí, stráví s produktem nějaký čas, přestává si spousty problémů všimnout, jelikož se s nimi naučí pracovat. Uživatelské testování použitelnosti tedy může připomenout, že ne každý přemýšlí stejně a poskytnout novou perspektivu. “Ale nakonec jsem přišel na to, že testování je jako když vás navštíví přátelé z jiného města. Když je budete provádět po vašem městě, nevyhnutelně uvidíte věci, kterých jste si obvykle nevšimli, protože jste na ně byli zvyklí. A současně si uvědomíte, že spousta věcí, které vám připadají jasné, nemusí být zřejmé všem”. [6] Tento citát perfektně ilustruje, proč je důležité uživatelské testování provádět.



Obrázek 2.1. Graf zobrazující poměr počtu uživatelů a nalezených chyb, převzato z [7].

Testování nemusí být velká věc, mnohem důležitější je provést ho včas a správně. Provést testování s jediným uživatelem je mnohem lepší než neprovést žádné a k odhalení většiny problémů stačí většinou pět až osm uživatelů, jak je ilustrováno na obrázku 2.1. I to nejhorší testování může poukázat na přehlížený problém. Nebo člověk, který není cílovým uživatelem aplikace, může přinést zajímavé poznatky. Cílem testování není něco dokázat, nebo vyvrátit, ale spíše poskytnout informace, které pomohou s rozhodnutím se pro nějaké řešení. [6]

2.4.1 Vývoj uživatelského testování

Tradiční uživatelské testování bylo velice náročné, jak časově, tak finančně. Probíhalo v testovacích místnostech, které byly vybaveny kamerami, falešným zrcadlem a mikrofony, aby se dal účastník testování dokonale sledovat. Testy byly také náročné na počet lidí, který je musel obsluhovat, aby bylo dosaženo co nejpřesnějších výsledků. Tyto skutečnosti vedly k tomu, že se uživatelské testy nedělaly příliš často.

V roce 1989 přišel Jakob Nielsen s referátem, kde ukazoval, že testování použitelnosti nemusí být tak náročné a k podobným výsledkům lze dojít se zlomkem nákladů a lidí.

Tato myšlenka se dá podle Stevena Krugera dále zjednodušit na takzvané testování bez smlouvy (*Do-it-yourself testing*), které veškeré nároky na testování snižuje. Rozdíly mezi oběma přístupy jsou zachyceny v tabulce 2.1. Kniha, ze které je tabulka převzata se zaměřuje převážně na webové stránky, ale princip je stejný i pro ostatní programy. Výsledkem testování bez smlouvy je větší dostupnost testování, které je možné provádět mnohem častěji a to zásadně napomáhá vývoji aplikace. [6]

	Tradiční testování	Testování bez smlouvy
Počet účastníků testu	Obvykle osm nebo více, aby to ospravedlnilo náklady na přípravu.	Tři nebo čtyři.
Náročnost náboru	Vybírejte pečlivě, aby lidé odpovídali cílové skupině uživatelů.	Vyhovovat bude skoro každý, kdo používá web.
Kde se bude testovat	Testovací laboratoř s pozorovací místností a falešným zrcadlem.	Kancelář nebo konferenční místnost.
Kdo bude testování provádět	Zkušený odborník na použitelnost.	Libovolná přiměřeně trpělivá osoba.
Potřeba plánování	Testy musí být naplánovány týdně dopředu, aby bylo možné rezervovat testovací laboratoř a bylo dost času na nábor.	Testy lze provést prakticky kdykoliv a naplánovat je stačí s krátkým předstihem.
Příprava	Návrh, projednání a přepracování testovacího protokolu.	Rozhodnout, co hodláte dokázat.
Co/kdy budete testovat?	Pokud nemáte obrovský rozpočet, dejte všechny problémy do jednoho pytle a testujte najednou těsně před dokončením serveru.	Provádějte menší testy nepřetržitě v průběhu celého vývojového procesu.
Náklady	5 000 až 15 000 dolarů (případně více).	Kolem 300 dolarů (50 až 100 dolarů jako plat každému uživateli a 20 dolarů za tři hodinové VHS kazety).
Co se stane potom	Za týden dostanete dvacetistránkovou zprávu, potom se sejde vývojový tým, aby rozhodl o případných změnách	Každý pozorovatel napíše stránku poznámek ještě v den testu. Vývojový tým může ve stejný den vyslechnout výsledky.

Tabulka 2.1. Tabulka porovnávající tradiční a zjednodušené testování převzata z [6].

■ 2.4.2 Co jsou testy použitelnosti

Testy použitelnosti jsou strukturované rozhovory, které se zaměřují na funkce produktu nebo jeho prototypu. Jejich hlavní součástí je seznam úkolů, který vykonává účastník testování, který je většinou zástupce cílové skupiny produktu. Po testování zanalyzují

výzkumníci pořízené nahrávky a poznámky a po větším množství těchto testů mezi sebou poznámky porovnají a vytvoří seznam problémů, které bude třeba řešit. Testy použitelnosti pomáhají odhalit, zda potenciální uživatel zachází s produktem, tak jak bylo zamýšleno a zda jeho fungování rozumí. Testy použitelnosti jsou v praxi často prováděny až před vydáním produktu, ale měly by být zařazeny do celého procesu vývoje. [8]

■ 2.4.3 Co a kdy testovat

Jak už bylo řečeno v sekci věnované iterativnímu designu, je důležité začít testovat co nejdříve a testování v každé vývojové fázi opakovat. Doporučuje se začít testovat už s nízkourovněnými (*low fidelity*) prototypy, jelikož umožňují včas identifikovat případné nedostatky návrhu a předejít tak zbytečným úpravám v závěrečné fázi procesu.

Testování použitelnosti zkoumá jak lidé vykonávají konkrétní úkoly. Nejsou tedy nejefektivnějším způsobem jak otestovat používání celého produktu najednou, ale je lepší zaměřit se na užší část. Testování finálních funkcionalit může pomoci spíše další verzi produktu, než té aktuální. Nejedná se o jednorázovou záležitost, ale o proces, kdy se v každé iteraci zaměřujeme na menší množinu funkcionalit. Existují čtyři hlavní typy testování použitelnosti: [8]

- výzkumné, které testuje nové koncepty.
- posuzující, které se věnuje funkcionalitám během implementace.
- porovnávací, které má za úkol rozhodnout, která varianta je lepší.
- validační, které ověřuje, že jednotlivé prvky splňují vytyčené standardy.

2.4.4 Jak vybrat účastníky testu

K výběru účastníků lze využít screener. Screener je nástroj, který umožňuje naplnit stanovené kvóty toho jak by měli uživatelé vypadat formou dotazníku. Tyto kvóty by měly být dostatečně flexibilní, aby bylo možné nalézt dostatečné množství účastníků pro testování. Ale neměly by být příliš obecné, aby zahrnovaly opravdu jen pravděpodobné uživatele produktu. Také lze určit důležitost jednotlivých kritérií. Lze rozlišit ta, která musí být dodržena vždy, a ta která nejsou tak důležitá, pokud potenciální uživatel splňuje vše ostatní. Co se týče ideálního počtu účastníků, tak záleží na tom jak dobře popisují cílovou skupinu. Je lepší mít menší počet reprezentativních uživatelů, než mnoho nerepresentativních. Ve většině případů stačí tři až čtyři uživatelé. První tři uživatelé často odhalí skoro všechny podstatné problémy. Daleko důležitější je provést více testovacích kol. Důvodem je to, že některé velké problémy zadrží uživatele tak, že si dalších problémů ani nevšimnou. Proto je lepší testovat s menším množstvím uživatelů, chyby opravit a pak testování opakovat. [6, 8]

2.4.5 Jak testování připravit

Důležitým krokem je výběr funkcionalit, které budou testovány. Je dobré zvolit přibližně pět nejdůležitějších. Důležitost je určena například tím, jak často je funkcionalita využívána, jak moc nová je, jestli již dříve způsobovala problémy a podobně. Potom, co jsou vybrány funkcionality k testování, je třeba připravit seznam úkolů, který co nejlépe pokryje možná využití těchto funkcionalit. Správné úkoly by měly být:

- smysluplné, mělo by to být něco s čím se uživatel běžně setkají.
- popsané jako úkoly, aby měl uživatel motivaci je plnit. Také není žádoucí popisovat postup krok za krokem.
- specifické, to zaručí konzistentnost mezi uživateli.

- v realistickém pořadí, aby simulovaly reálné použití.
- rozumně dlouhé.

Někdy může být vhodné nechat účastníka vymyslet si vlastní úkol, aby měl větší motivaci ho splnit a používal produkt s jasným cílem. Po zvolení úkolů následuje jejich sepsání ve formě skriptu (*session guide*). Tento dokument má většinou tři části: Instrukce a úvodní otázky, úkoly a závěr. Instrukce obsahují základní informace o průběhu testování, slouží k uvolnění a připravení účastníka. Například je dobré uvést, že problémy, na které účastník testování narazí nejsou jeho chyba, ale pravděpodobně chyba produktu. Další dobrou praktikou je upozornit účastníka, aby své postupy a myšlenkové pochody popisoval nahlas. Úvodní dotazník slouží k doplnění dodatečných informací o účastníkovi. A hlavní funkcí závěru je poskytnout prostor pro otázky které bude mít účastník a získat jeho zpětnou vazbu. [8]

■ 2.4.6 Průběh testování

Hlavním cílem testování je, aby se účastníci testování chovali co nejpřirozeněji a všechny části testování by tomu tedy měly napomáhat. Místo, kde se bude testování konat, by mělo reprezentovat prostor, kde budou lidé produkt používat. Naskýtají se tedy dvě možnosti provedení testu. Navštívení účastníka přímo u něj doma, případně v práci, a nebo testování ve vlastním zařízení. V případě návštěvy u uživatele není třeba prostředí nijak upravovat. Naopak při využití nějakého jiného místa je třeba simulovat cílové prostředí. Pokud je například produkt používán doma, je dobré, aby prostor připomínal pracovnu nebo obývací pokoj. Tento efekt nemusí být perfektní, ale pár doplňků, jako například květina, způsobí, že se bude účastník testování cítit více jako doma, místo jako v laboratoři a jeho chování bude přirozenější. Během testování je s účastníkem v místnosti moderátor. Hlavním úkolem moderátora je, aby se účastník během testování cítil pohodlně, sledovat jeho chování, nebo se doptávat na dodatečné informace a dělat si poznámky. Do testování použitelnosti mohou být zapojeni také vývojáři a manažeři. Je to nejlepší způsob, jak jim předat informace z testování, jelikož sami vidí, jak je jejich produkt používán a jestli je to v souladu s jejich očekáváním. Tito lidé jsou během testování ve vedlejší místnosti a testování pozorují za pomoci kamer a mikrofónů nebo speciálního falešného zrcadla. [8]

■ 2.4.7 Zpracování výsledků

Po dokončení testování je třeba nasbíraná data projít a analyzovat. Zjistit za jakých okolností docházelo k jakým chybám, které chyby byly časté, co je pravděpodobně způsobilo a podobně. Testování použitelnosti je z velké části kvalitativním testováním a při malém počtu participantů se nevyplatí dělat z nasbíraných dat statistiky. Pro získání kvantitativních dat často stačí jednoduchá škála k určení, jak dlouho trvalo lidem úkol splnit:

- 0 . Nepovedlo se
- 1 . Povedlo se pomalu a složitým způsobem
- 2 . Povedlo se pomaleji
- 3 . Povedlo se rychle

Tato data se dají zprůměrovat pro každý úkol a získat finální ohodnocení jednotlivých úkonů. Druhým krokem analýzy výsledků je seskupit pozorování dle jednotlivých částí programu a najít podobnosti mezi účastníky. Problémy, které se objevily jen u jednoho člověka, nebudou tak důležité jako ty, co potkali každého. Závěrem je třeba všechna data zpracovat ve formě reportu. [8]

Kapitola 3

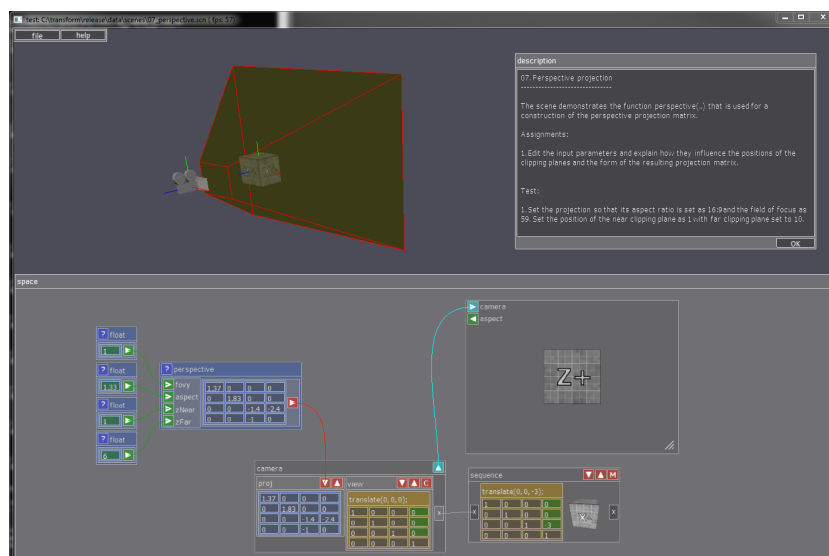
Úvodní testování použitelnosti I3T

Jelikož vývoj aplikace probíhal střídavě v rámci různých bakalářských a diplomových prací, nebyla použitelnost řešena nijak systematicky. Někteří autoři prací otestovali s uživateli části aplikace, kterým se věnovali a tam jejich práce skončila. Dala jsem si tedy za úkol zaměřit se na použitelnost aplikace více do hloubky.

Aplikace již v minulosti prošla pár uživatelskými testováními, ze kterých je možné vycházet. Přesto jsem se rozhodla na úvod provést vlastní uživatelské testování, a to ze dvou hlavních důvodů. Prvním z nich je skutečnost, že od posledního testování uplynuly přibližně dva roky, během nichž došlo v aplikaci k zásadním změnám. Některé závěry z původního testování tak již nemusí být relevantní. Druhým důvodem je význam osobní zkušenosti získané provedením testování. Při pouhém čtení výsledků testování může snadno docházet k podcenění nebo přehlížení některých problémů. Ty je možné si naplno uvědomit až při přímém pozorování interakce uživatele s rozhraním.

3.1 Předchozí testování

Je vhodné zmínit, že již v roce 2017 proběhlo uživatelské testování původní verze programu (obrázek 3.1) vytvořené Michalem Foltou [1]. Testování poukázalo spíše na technické nedostatky, než na problémy v použitelnosti. Jedním z výsledků, bylo oddělení modelu od sekvence, které se dříve vyskytovaly společně. Testování také poukazuje na problém s přidáváním transformací do sekvence, kdy se uživatelé snažili hledat uzly transformací přímo v sekvenci místo v přidávacím menu. Nicméně vzhledem k rozsáhlým změnám v aplikaci, které od té doby proběhly, není většina dalších závěrů testování pro mou práci relevantní.



Obrázek 3.1. Vzhled původního rozhraní ve Foltově programu [1].

Použitelnosti se také věnoval ve své práci Vít Zadina v roce 2019 [9]. Jeho práce se z velké části soustředila na klávesové zkratky. Prováděl kvantitativní výzkum formou dotazníku, kdy pro každou důležitou funkci programu navrhl několik klávesových zkratk a nechal účastníky zvolit tu nejlepší. V druhé části své práce vytvářel prototyp rozhraní v programu *Axure*, který testoval i využitím uživatelského testování. Z tohoto testování vzešlo několik závěrů, které jsou pro mojí práci významné.

- Název sekvence není intuitivní a došlo k přejmenování na Blok matic, jelikož název sekvence nebyl pro uživatele dostatečně intuitivní. Tato změna se v programu nezachovala i přes toto zjištění.
- Nová scéna by měla být prázdná namísto otevření se s již vloženou sekvencí a modelem.
- Na základě testování bylo umožněno vkládat transformaci přímo do sekvence pravým kliknutím na oblast uzlu a zvolením transformace z nabídky. Tato funkce není v nové verzi dostupná, ale jedná se o jedno z možných řešení pro navedení uživatele k tomu, že je třeba využít sekvence k napojení modelu.
- Během testování vznikl nápad, umožnit vkládat modely přímo do scény (*scene view*).

V roce 2022 hodnotili studenti Vysoké školy ekonomické I3T z hlediska uživatelské přívětivosti. Testování provedené studenty VŠE testovalo použitelnost aplikace za pomoci tutoriálů. Nejprve měl účastník testování za úkol projít prvním tutoriálem Začínáme s I3T a tutoriálem Základy transformací a práce s nimi. Potom bylo participantovi zadáno vytvořit v programu krychli a aplikovat na ní transformaci škálování a translace. A posledním úkolem bylo vytvořit scénu, v ní vytvořit model a ten otočit o 30°. Po každém z těchto úkolů a také na závěr testování vyplnil každý z účastníků krátký dotazník. Výsledkem tohoto testování bylo několik zjištění týkajících se většinou tutoriálů. Pro mojí práci jsou tedy relevantní pouze čtyři z nich:

- Menu jsou složitá na pochopení. Účastníkům testování chybělo vizuální rozdělení jednotlivých prvků menu a měli problém s pochopením co jednotlivé položky znamenají.
- Maticové násobení logicky nepatří k dalším typům napojení. Na rozdíl od ostatních datových typů napojení jako je matice nebo vektor je maticové násobení operace. Dle participantů by měl program tuto skutečnost reflektovat.
- Uživateli se nepodařilo objevit, že lze pravým kliknutím zobrazit často používané hodnoty v maticích.
- Po přejmenování uzlu není jasné, co představuje. Bylo by dobré, aby byl typ uzlu nějakým způsobem stále dostupný.

3.2 Úvodní testování

Rozhodla jsem se začít provedením souhrnného testování použitelnosti celé pracovní plochy *workspace*. Některé chyby a nedostatky byly zřejmé již od začátku, avšak mnoho problémů si uživatel, který s aplikací pracuje pravidelně, nemusí vůbec uvědomovat. Často se také stává, že postupy, které se zkušenému uživateli jeví jako samozřejmé, mohou být pro nové uživatele neintuitivní nebo obtížné na pochopení. Cílem testování bylo zjistit, jakým způsobem s aplikací pracují uživatelé bez předchozích zkušeností a identifikovat oblasti, které jim působí největší potíže.

Jelikož je tato oblast programu velmi obsáhlá, zaměřila jsem se na úkony, které uživatelé provádí nejčastěji. Z uzlů, jsem se soustředila převážně na model, sekvenci, transformace, kameru a obrazovku.

Úvodní testování probíhalo na verzi programu v2.0.0-rc3 z listopadu 2024.

3.2.1 Metodologie

Účastníci testování měli za úkol vyřešit celkem tři úlohy. Každá z nich simulovala situaci, se kterou se lze běžně setkat v rámci výuky předmětu Programování grafiky, aby účastníci měli jasnou a smysluplnou motivaci k jejich splnění. Úlohy byly navrženy tak, aby pokrývaly většinu základních funkcionalit *workspace*, se kterými se běžný uživatel může setkat:

- Vytvořit dva modely, změnit jejich barvu a na oba aplikovat dvě shodné transformace, ale v jiném pořadí.
- Přidat do scény kameru, tu doplnit o transformace pohledu a projekce (view a projection), připojit k ní obrazovku (*screen*) a nastavit parametry kamery tak, aby byly předem vložené modely viditelné na obrazovce.
- Vyřešit transformační úlohu. Úkolem bylo převést objekt z jednoho souřadnicového systému do jiného.

Při provádění testování s uživateli je k výběru participantů testování často využíván nástroj jménem screener. Jedná se o dotazník, který slouží k určení toho, kdo je pro testování vhodný a kdo ne. [8] Já jsem se v tomto úvodním testování zaměřila na studenty, kteří již absolvovali předmět Programování grafiky a mají o transformacích v počítačové grafice povědomí, ale s programem I3T zatím nepracovali. Screener by byl v tomto případě zbytečným krokem, jelikož mi stačilo oslovit studenty se zaměřením Počítačové hry a grafika, kteří předmět absolvovali v letním semestru 2023. Touto dobou byl nástroj I3T využíván, jen jako vizualizační nástroj na přednáškách, ale sami studenti ho nepoužívali. Tato volba umožnila pozorovat, jak s aplikací zacházejí uživatelé, kteří vědí, co chtějí udělat, ale nevědí, jak toho dosáhnout.

Testování proběhlo v budově školy v prázdných učebnách nebo studovnách v pracovním týdnu 18. až 22. listopadu. Každému účastníkovi byl k dispozici notebook se spuštěným programem a papír se zadáním úkolů. Během práce měl účastník za úkol nahlas popisovat své myšlenkové postupy a kroky, které v aplikaci prováděl.

Tyto postřehy jsem si průběžně zaznamenávala a následně je seskupila do kategorií podle toho, čeho se týkaly. Identifikovala jsem oblasti, které byly problematické pro více účastníků testování. A vytvořila seznam problémů použitelnosti, kterým bylo třeba se věnovat.

3.2.2 Výsledky

Během testování se ukázalo, že existuje několik zásadních problémů, se kterými se potýkali všichni účastníci:

- **Přidávání objektů do workspace.** Hledali přidávací menu v horní nabídce, klikali do scény nebo procházeli všechna tlačítka. Až po delší době, nebo vůbec, je napadlo kliknout pravým tlačítkem myši do *workspace*, což je v současné verzi programu jediný způsob, jak otevřít přidávací menu.
- **Aplikování transformace na model.** Matice transformace, kterou lze do *workspace* vložit, nemá sama o sobě žádné vstupy ani výstupy a nelze ji tedy připojit přímo k modelu. K napojení transformace na model je třeba ji nejprve vložit do sekvence. Tento postup, ale není pro nového uživatele zřejmý. Ve dvou případech účastníci dokonce zcela obešli použití transformační matice, místo toho využili operátor transformace jako vstup pro sekvenci a tu následně připojili k modelu. Tento přístup sice funguje, ale je zbytečně složitý a není žádoucí, aby ho uživatelé využívali.
- **Chybové hlášky.** Program v některých případech zobrazí uživateli chybovou hlášku informující o tom, že určité spojení nelze vytvořit, ale uživatel se nedozví, proč. To

vedlo k frustraci, protože nebylo jasné, co přesně chybu způsobilo, nebo jak jinak má uživatel postupovat.

- **Implicitní napojení kamery na kořeny uzlů.** Uživatelé si nebyli jisti, zda se jedná o skutečné zapojení a zda ho mají nějak upravit nebo odpojit.
- **Úprava parametrů modelu.** Uživatelé měli za úkol změnit barvu modelu. Správně předpokládali, že tuto možnost najdou v menu modelu, ale nepodařilo se jim ho nalézt. Důvodem bylo to, že pro otevření menu modelu je nutné kliknout pravým tlačítkem na horní část uzlu, nikoliv na samotný model nebo jeho vizuální reprezentaci v dolní části uzlu.

Kromě těchto hlavních problémů z testování vyplynuly i další menší poznatky, které mohou vést ke zlepšení aplikace:

- **Malá velikost vstupů a výstupů.** Stávalo se, že propojit uzly se podařilo participantovi až na několikátý pokus.
- **Označení typů napojení.** V současné verzi jsou si všechny typy vstupů a výstupů velmi podobné, což vedlo u uživatelů ke zmatení mezi kopírováním hodnoty a násobením matic.
- **Zamčené hodnoty matice.** Některé vložené matice jsou ve výchozím zobrazení v zamčené podobě. To participanty mátló, jelikož netušili, proč nemohou měnit jejich hodnoty.
- **Naležitelnost funkcionalit.** Některé funkce, které uživatelé chtěli využít, se jim nepodařilo objevit a díky tomu se domnívali, že neexistují. Například možnost resetovat hodnoty nebo přepínat režim zobrazení transformačních matic, jako je možné například u lookAt matice.

Kapitola 4

Nalezené problémy

Na základě provedeného úvodního testování použitelnosti, vlastního pozorování a konzultace s týmem podílejícím se na vývoji aplikace jsem vytvořila seznam problémů a nedokonalostí aplikace I3T, které bylo třeba, v rámci zlepšení její použitelnosti, vyřešit. V následující sekci jsou popsány nalezené problémy, rozdělené do kategorií podle toho, které části pracovní plochy *workspace* se týkají. V této části je rozebrána verze aplikace z listopadu 2024, v2.0.0-rc3, na které probíhalo úvodní testování.

4.1 Celý pracovní prostor

V této verzi je možné se v rámci *workspace* pohybovat pouze myší, jelikož je k pohybu využíváno prostřední tlačítko, což omezuje uživatele využívající notebook bez připojené myši. Druhým zásadním problémem je velikost písma – text je velmi malý a obtížně čitelný, což dále ztěžuje orientaci v prostředí aplikace. Na monitorech s vysokým rozlišením je pak aplikace téměř nečitelná, což výrazně omezuje její použitelnost.

4.2 Menu

Největším problémem menu je jeho nízká dohledatelnost pro nové uživatele. V této verzi se menu otevírá pouze kliknutím pravým tlačítkem myši do prostoru *workspace*, což není intuitivní. Navíc samotné rozhraní neposkytuje žádnou vizuální nápovědu, která by uživatele nasměrovala ke správnému způsobu otevření menu.

Dalším problémem je samotné rozmístění položek v menu. Transformace jsou umístěny přímo nad operátory, což často vede k jejich záměně. Pořadí položek odpovídá pořadí jejich implementace, nikoli potřebám uživatele, což prodlužuje dobu potřebnou k orientaci a vyhledání požadované funkce.

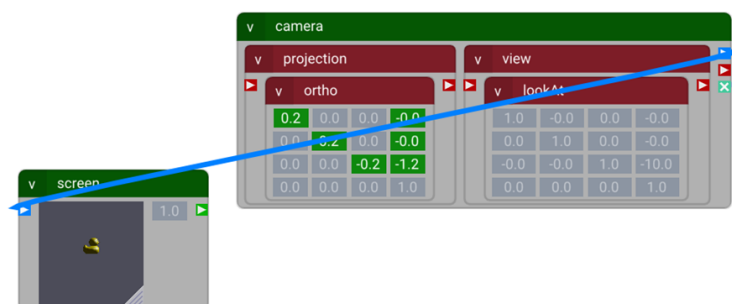
Posledním nedostatkem je otevírání částí menu mimo obrazovku v případě, že uživatel klikne příliš blízko jejího okraje. Tento jev snižuje efektivitu vyhledávání, protože uživatel musí menu opakovaně otevírat, pokud se požadovaná položka dostane mimo viditelnou oblast. U nových uživatelů může tento problém navíc způsobit zmatení, jelikož mohou nabýt dojmu, že požadovaná možnost v nabídce vůbec neexistuje.

4.3 Dráty

Stává se, že dráty vedoucí z uzlů překrývají ostatní uzly ve *workspace*, což negativně ovlivňuje přehlednost celého rozhraní, jak je viditelné na obrázku 4.1. Dále v této verzi aplikace chybí možnost odstranit dráty jejich označením a stisknutím klávesy „Delete“, jak je tomu u uzlů. Je možné odstraňovat dráty pouze dvěma způsoby:

- Kliknutím na koncový vstup, ke kterému je drát připojen. Tento způsob vede ke zmatení, pokud uživatel klikl na vstup omylem a drát zmizel.

- kliknutím pravým tlačítkem myši přímo na drát a zvolením možnosti „Delete“ z kontextového menu. Tento způsob ale není pohodlný, jelikož jsou dráty tenké a je obtížné je přesně zasáhnout kurzorem myši. Navíc odstranění objektu pomocí klávesy „Delete“ patří k zažitým standardům v podobných aplikacích.

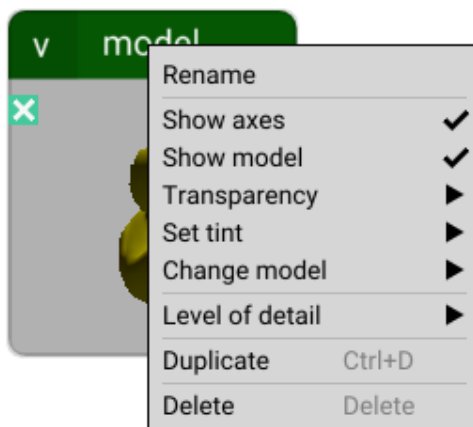


Obrázek 4.1. Vykreslení drátu přes uzly.

4.4 Uzly

Jedním ze zjištěných problémů bylo, že uživatelé nenalezli kontextová menu jednotlivých uzlů, jelikož neexistuje žádný vizuální indikátor jejich přítomnosti.

Další problém souvisí s orientací v těchto menu. Některá z nich obsahují velké množství možností pro přizpůsobení, které běžní uživatelé zpravidla nevyužívají. Důsledkem toho je, že se mezi těmito volbami ztrácí důležité funkce, které uživatelé hledají, nebo které by je mohli zajímat. Příkladem je kontextové menu modelu zachycené na obrázku 4.2.

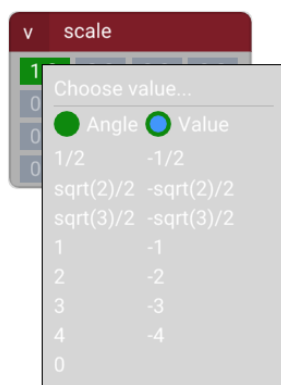


Obrázek 4.2. Kontextové menu modelu ve verzi v2.0.0-rc3.

Další nejasnosti vznikají při rozlišování různých typů napojení (*pinů*) mezi uzly. Speciálním druhem napojení je maticové násobení, které je v této verzi umístěno v horní části uzlu. Takové umístění je pro uživatele neintuitivní, jelikož v matematice bývá maticové násobení umístěno ve středu mezi maticemi. V důsledku toho uživatelé vnímají tento typ napojení jen jako další druh vstupu nebo výstupu a tím přicházejí o důležitou souvislost. Tato skutečnost ovlivňuje porozumění celé vizualizaci a pochopení toho co

se jim program snaží přiblížit. Druhý problém se týká velikosti napojení. Napojení jsou příliš malá, což znesnadňuje jejich uchopení a manipulaci s nimi. Uživatelé často potřebují několik pokusů, než se jim podaří napojení uchopit, což vede ke zbytečné frustraci a snižuje efektivitu práce s aplikací.

Všeobecně se u uzlů objevuje problém se špatnou naležitelností dostupných funkcí a možností interakce. Například možnost změnit hodnotu matice je dostupná kliknutím myši na hodnotu a následným pohybem doleva nebo doprava ke snížení či zvýšení hodnoty. Často používané hodnoty lze zobrazit kliknutím pravým tlačítkem myši na pole s hodnotou. Menu s výběrem hodnot je viditelné na obrázku 4.3, obrázek také poukazuje na problém špatné čitelnosti tohoto menu ve světlém režimu. Tyto funkce však nejsou nijak vizuálně indikovány. Uživatelé je často neobjeví a vkládají hodnoty ručně pomocí klávesnice, čímž přicházejí o pohodlnější způsob zadávání hodnot.



Obrázek 4.3. Menu pro výběr často používaných hodnot.

Dalším příkladem je nastavení režimu zobrazení uzlu. Všechny uzly umožňují přepínání mezi zjednodušeným režimem (zobrazení pouze horní lišty uzlu) a plným režimem. Některé uzly nabízejí i další specifické režimy, například uzly transformačních matic obsahují režim zadávání parametrů, který umožňuje zadávat pouze vektor nebo úhel namísto celé matice. Tento režim je velmi užitečný pro rychlé zadání konkrétní hodnoty bez nutnosti sestavovat celou matici. Proto by bylo dobré, aby byl režim pro uživatele jednoduše naležitelný.

Některé uzly navíc podporují možnost zamykání hodnot. Například u matice translace jsou zamčené všechny hodnoty kromě horních tří v pravém sloupci, protože pouze tyto hodnoty ovlivňují posunutí objektu. Specifickým případem je matice typu `lookAt`, která je při vložení do scény automaticky v uzamčeném režimu, kde žádné hodnoty nelze přímo upravovat. Důvodem je, že jakákoliv ruční změna by mohla porušit pravidla, která musí tato matice splňovat. Parametry této matice se proto upravují výhradně v režimu zadávání parametrů pomocí vektorů `up`, `center` a `eye`, ze kterých je výsledná matice spočítána. Automatické zamčení však působí na uživatele matoucím dojmem a vyvolává pocit, že s maticí nelze vůbec pracovat.

4.5 Úhly

V programu je hojně využívána hodnota typu `float`. Někdy jako libovolná reálná hodnota nebo hodnota z intervalu a někdy jako úhel. Tato nejednoznačnost může být pro

uživatele matoucí. Také způsob zadávání hodnot úhlů není ideální. Například při nastavování rotačních úhlů by bylo přirozenější pracovat se stupni namísto desetinných hodnot, které odpovídají radiánům.

4.6 Sekvence

Největším problémem sekvencí je, že uživatelům není zřejmé, že do nich lze vkládat transformace a že sekvence slouží k napojení těchto transformací na model. A program o tom sám o sobě nijak neinformuje.

Dalším problémem je speciální typ výstupu, který sekvence poskytují. Tento výstup obsahuje hodnotu násobku všech napojených transformací směrem doleva až k počátečnímu uzlu (kořeni). Tento výstup však není nijak odlišen od standardního výstupu a uživatel tedy nepozná jaký je mezi výstupy rozdíl jinak než metodou pokus-omyl. Uživatel se také může chybně domnívat, že oba výstupy mají stejnou funkci.

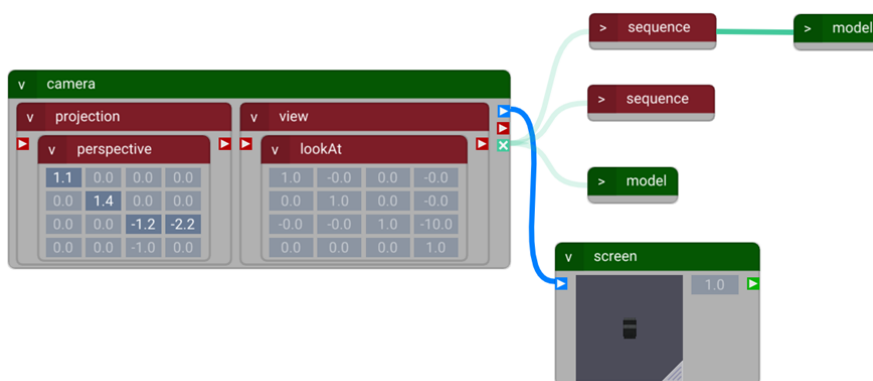
4.7 Krokování (*tracking*)

Sekvence umožňuje také krokování (*tracking*) transformace, ale tato funkcionality není dosud zcela vyřešena. Aby krokování fungovalo správně, musí být na vybranou sekvenci napojen model. Možnost krokování je však zobrazena u všech sekvencí, přestože u těch, které nemají připojený model, po spuštění neproběhne žádná akce. Dalším problémem je, že při sledování transformace je výsledná pozice modelu zobrazená zcela neprůhledně. To způsobuje problém v případě, že je původní model **uvnitř** finálního modelu, protože proces krokování není v tomto případě vůbec viditelný.

4.8 Kamera

Prvním nedostatkem u uzlu kamery je umístění napojení na obrazovku *screen*, které je v současnosti na levé straně uzlu (obrázek 4.4). Toto uspořádání však není intuitivní vzhledem k tomu, jak sekvence transformací aplikovaných na model odpovídá reálnému řetězci operací.

Dalším matoucím aspektem je pomyslné napojení kamery na kořenové uzly transformací (obrázek 4.4). Toto spojení může být pro uživatele zavádějící, protože není zřejmé, zda je kamera skutečně připojena nebo ne. Zároveň výstup kamery určený pro maticové násobení nelze nikam připojit, což působí jako chyba.



Obrázek 4.4. Zapojení uzlu obrazovky do kamery a implicitní zapojení kamery.

4.9 Model

Výchozím modelem přidáním do scény je krychle. Řada uživatelů však ani neví, že vzhled modelu lze změnit, protože tato možnost je skrytá v kontextovém menu. Prostor pro zlepšení je také v nastavení barvy modelu, které je v současné podobě zbytečně komplikované. Provádí se přes volbu tónování (*tint*) a následné nastavení intenzity zbarvení.

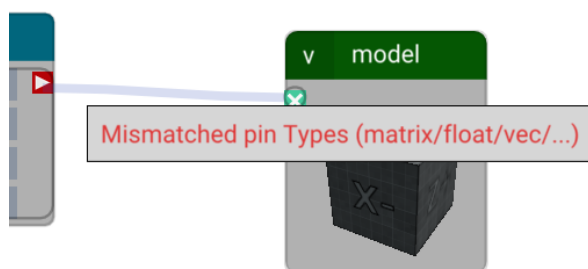
4.10 Cyklus

Cyklus je sám o sobě velký uzel a zabírá v pracovní ploše hodně místa. Také je nepřehledný, kvůli velkému množství vstupů a výstupů, a náročný na pochopení pro nové uživatele.

4.11 Zpracování chyb (*error handling*)

Během používání programu se může uživatel dostat do situace, kdy provede chybnou operaci, na kterou ho program upozorní. A to konkrétně ve dvou hlavních případech:

- Nesprávné zapojení uzlů. Program uživateli oznámí nesoulad typů pinů (obrázek 4.5), nicméně nespécifikuje, o jaké typy se konkrétně jedná. Chybová hláška je navíc pro všechna zapojení shodná, což uživateli znemožňuje jednoduše zjistit příčinu problému.



Obrázek 4.5. Chybová hláška při neshodě typů napojení.

- Nastavení chybné hodnoty v transformaci. Při zadání nevalidních hodnot se v horní části rozhraní zobrazí chybová ikona (obrázek 4.6), avšak bez dalšího vysvětlení proč k chybě došlo. Nejčastější příčinou je, že matice přestane splňovat podmínky dané transformace. Například pokud se na místě, kde musí být nula, objeví jiná hodnota. Ikona chybového hlášení na červeném pozadí působí nevýrazně a dostatečně neupozorňuje uživatele na to, že došlo k chybě.

v	lookAt		
1.0	-0.0	0.0	-0.0
0.0	1.0	0.0	-0.0
-0.0	-0.0	4.0	-10.0
0.0	0.0	0.0	1.0

Obrázek 4.6. Chybné hodnoty uvnitř uzlu lookAt transformace.

Kapitola 5

Návrh změn

Po identifikaci jednotlivých nedostatků a problémových oblastí jsem se věnovala návrhu jejich možných řešení a zároveň vymýšlela další návrhy pro zlepšení celkové použitelnosti aplikace. Možné varianty řešení jsem navrhovala formou skic a dále je konzultovala s týmem, a to jak z hlediska vizuálního zpracování, tak z pohledu technické proveditelnosti. Výsledné návrhy jsem následně testovala ve třech iteracích s uživateli (kapitola 6), s cílem ověřit, zda skutečně řeší identifikované problémy, zda stávající návrh vyžaduje další úpravy, případně zda je nutné vymyslet alternativní řešení.

5.1 Konzultace s týmem

Na návrhu rozhraní pracovní plochy *workspace* se podílel také Dan Rakušan [10], který se v rámci semestrálního projektu věnoval refaktoringu kódu, který implementuje uživatelské rozhraní. Díky této práci měl dobrý přehled o tom, které návrhy jsou technicky proveditelné a jak náročná by jejich realizace mohla být. Tuto situaci jsme využili ke spolupráci na vylepšení programu a společnými konzultacemi jsme dále rozvíjeli existující nápady. Kolega Rakušan rovněž přispěl několika vlastními návrhy, na které jsem mohla navázat, a které případně sloužily jako alternativy k mým řešením. Tyto návrhy jsou uvedeny u jednotlivých konkrétních případů níže.

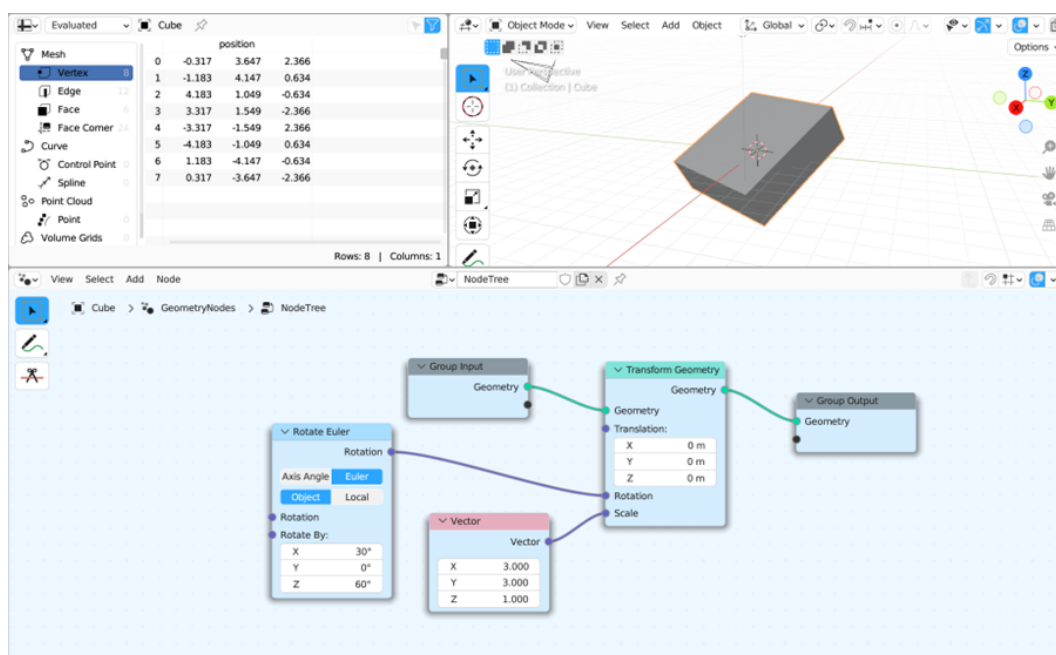
Tento přístup se ukázal jako velmi efektivní, zejména díky možnosti získat zpětnou vazbu od někoho, kdo má s daným systémem hlubší zkušenosti a mohl nabídnout vlastní pohled na strukturu i fungování rozhraní. Od druhé etapy testování jsme navíc důsledněji uplatňovali princip vývoje orientovaného na uživatele (User-centered development). Tento přístup spočíval v tom, že vybrané návrhy byly implementovány kolegou Rakušanem, následně jsem je otestovala s uživateli a zpracovala výsledky. Na základě těchto poznatků vznikaly nové návrhy, případně se ty stávající dále upravovaly a celý proces se opakoval. Tento proces je podrobněji zachycen v kapitole ref[testing].

Konzultace návrhů probíhaly také s vedoucím práce, který zastupuje skupinu expertních uživatelů a využívá program ve výuce jako vizualizační nástroj. Díky těmto konzultacím jsem získala cenné informace o konkrétních příkladech použití programu (*use-cases*), které bylo důležité zohlednit při návrhovém procesu. Jedním z příkladů je tzv. krokování, které slouží ke zdůraznění způsobu aplikace transformace na model – buď zprava, kdy je transformován samotný model, nebo zleva, kdy je transformována soustava souřadnic, ve které je model vykreslen. Bylo proto nezbytné, aby navrhované řešení zohledňovalo i tento scénář. Dále jsem se díky těmto konzultacím seznámila s využitím některých funkcí a klávesových zkratk, které hrají důležitou roli při efektivní práci s programem. Získala jsem také zpětnou vazbu na své návrhy od uživatele s hlubokou znalostí prostředí a dlouhodobou zkušeností, což mi umožnilo návrhy dále upravit tak, aby neomezovaly či nekomplikovaly práci pokročilým uživatelům a respektovaly jejich osvědčené postupy.

5.2 Rešerše uživatelských rozhraní podobných programů

Při návrhu uživatelského rozhraní jsem čerpala inspiraci z aplikací, které také využívají uzlový systém. Cílem bylo analyzovat, jakým způsobem tato rozhraní řeší interakci s uzly a jak přistupují k problémům, podobným těm, s nimiž jsem se setkala během návrhového procesu. Součástí rešerše bylo prozkoumat vizuální ztvárnění uzlů, způsob jejich přidávání, propojování a úprav. Zkoumala jsem jak tyto programy řeší zpracování chyb a jaká je jejich celková uživatelská přívětivost.

Prvním zástupcem je Blender¹ – open-source software určený především pro 3D modelování, tvorbu scén, animací a vizualizací. Systém uzlů využívá na několika místech, zejména v nástroji Geometry Nodes (obrázek 5.1), který slouží k detailní úpravě geometrie objektů. Umožňuje nejen základní transformace, jako je škálování, rotace a translace, ale i velké množství mnohem složitějších operací. Jedná se o pokročilý modelovací nástroj, zatímco I3T je navrženo jako jednodušší a přístupnější pro snadnější pochopení základních konceptů transformací. Systém uzlů v Blenderu je dále využíván i pro práci se shadery, přičemž způsob manipulace s uzly je velmi podobný jako u Geometry Nodes.



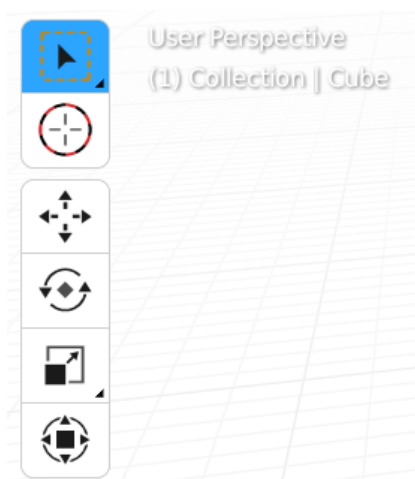
Obrázek 5.1. Vzhled pracovní plochy pro manipulaci s Geometry Nodes v programu Blender.

Uživatel může uzly vzájemně propojovat a upravovat jejich parametry. Přidávání nových uzlů je možné několika způsoby: prostřednictvím nabídky v horní liště, pravým kliknutím do editoru, nebo tažením drátu z výstupu/vstupu uzlu a následným vyhledáním požadovaného objektu, což je při velkém množství uzlů výhodné. Menu pro přidání v horní liště okna je vždy viditelné, tedy umožňuje snadné nalezení způsobu přidání, což je přístup, který by bylo možné využít i v I3T. Inspirací je také způsob zobrazení sbalených uzlů, u kterých se i ve sbalené podobě zobrazují všechna možná napojení. Tento

¹ <https://www.blender.org/>

přístup však není ideální u složitějších uzlů s velkým množstvím vstupů a výstupů, protože i ve sbaleném režimu zabírají značné množství prostoru. Jednotlivá napojení jsou v editoru reprezentována barevně odlišenými kolečky, jejichž barva odpovídá typu propojení. Jsou navíc mírně vysunutá z těla uzlu, čímž napomáhají vizuálnímu rozlišení mezi vstupy a výstupy.

Odpojování drátů je realizováno potažením za koncový uzel, což je jednoduché a uživatelsky přívětivé řešení. Pro usnadnění práce s uzly může uživatel vložit nový uzel přímo na stávající drát, pokud jsou jeho vstupy a výstupy kompatibilní. V případě chybného propojení systém upozorňuje uživatele zčervenáním spojovacího drátu. Tento způsob signalizace se jeví jako málo názorný a zároveň nutí uživatele provádět dodatečný krok, neboť vyžaduje manuální odpojení chybně propojeného drátu. Další užitečnou funkcionalitou je možnost seskupování uzlů do bloků, které napomáhají orientaci v komplexních grafech. Tyto bloky lze navíc transformovat do jediného uzlu, čímž se dále zvyšuje přehlednost a modularita. Tento princip by mohl být inspirativní i pro systém I3T, zejména v případě složitějších scén, pro organizaci pracovního prostoru. Kromě uzlového editoru může být inspirací i ikonové menu v levé části pracovní plochy (obrázek 5.2), které slouží k vkládání objektů a jejich transformaci. Toto menu přehledně představuje dostupné nástroje a přirozeně vede uživatele k první interakci se systémem.



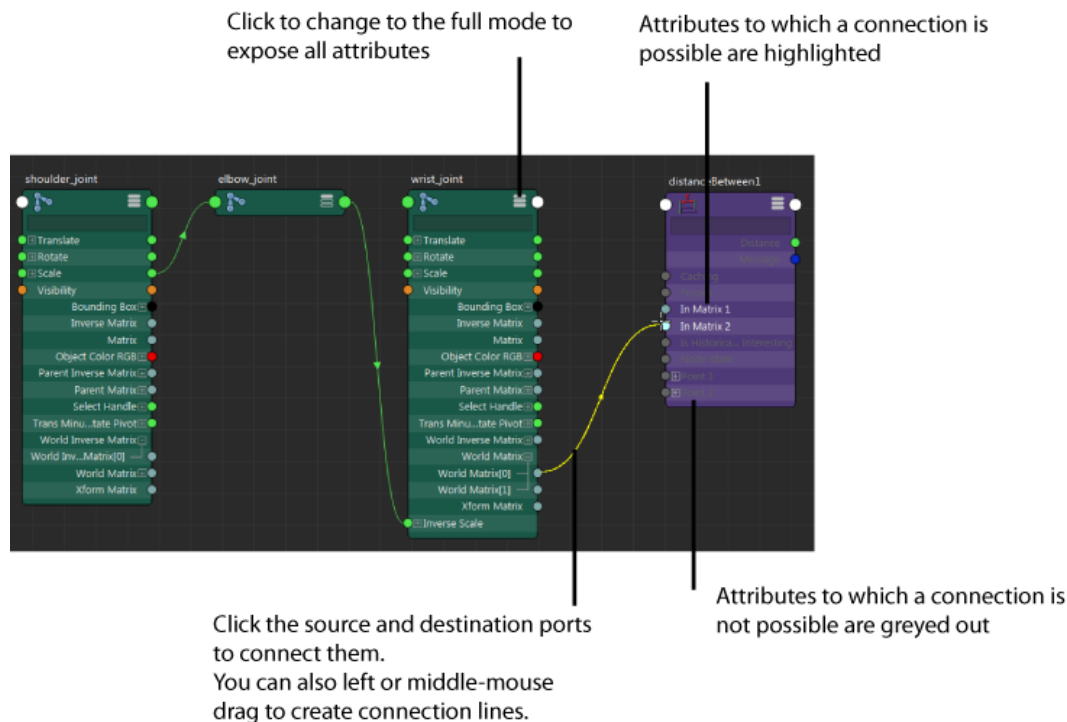
Obrázek 5.2. Vzhled ikonového menu v programu Blender.

Druhým programem, jehož uživatelské rozhraní bylo předmětem analýzy, je Autodesk Maya² – profesionální software určený pro 3D modelování, tvorbu animací a vizuálních efektů. Podobně jako Blender zahrnuje rozhraní pro manipulaci s uzly (obrázek 5.3), například při vytváření animace kostry.

Autodesk Maya nabízí zajímavý systém pro přepínání režimů jednotlivých uzlů, který by mohl sloužit jako inspirace i pro návrh uzlů v I3T. Každý uzel má v pravé horní části tři horizontální pruhy, které se postupně vyplňují v závislosti na aktuálním režimu. Podobně jako Blender, i Maya umožňuje efektivní vyhledávání uzlů podle názvu.

Odpojování drátů je v editoru řešeno několika způsoby: tažením spoje pryč, jeho výběrem a stisknutím klávesy „Delete“, případně kombinací „Alt + Shift“ a tažením přes více spojů současně. Tato funkce je sice užitečná, ale vzhledem k důrazu na jednoduchost uživatelského rozhraní by mohla být pro prostředí I3T zbytečně komplikovaná. Vzhled napojení je velmi podobný tomu v programu Blender – barevně rozlišené vstupy

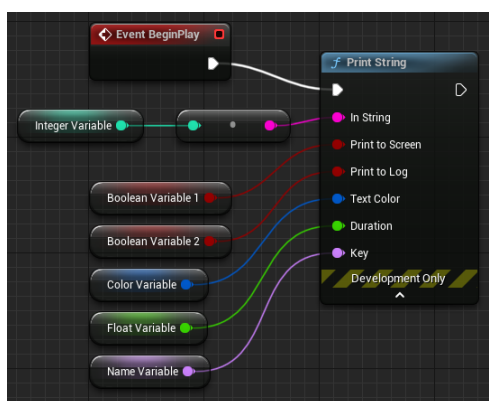
² <https://www.autodesk.com/cz/products/maya/overview>



Obrázek 5.3. Vzhled uzlů v programu Maya převzato z [11].

a výstupy s kruhovými konektory. Maya navíc implementuje i upozornění na možná zapojení. Při tažení drátu nepovolené vstupy dočasně zešednou, což efektivně napomáhá uživatelské orientaci a snižuje pravděpodobnost chybného propojení.

Posledním programem, jehož uživatelské rozhraní bylo předmětem zkoumání, je Unreal Engine³. Tento nástroj využívá uzlový systém na několika místech, zejména v rámci vizuálního skriptovacího nástroje Blueprint a editoru materiálů. V případě systému Blueprint se jedná o vizuální programování, které je sice funkčně odlišné od systému I3T, nicméně obsahuje řadu prvků, které mohou sloužit jako inspirace.



Obrázek 5.4. Vzhled uzlů nástroje Blueprint v Unreal Engine, převzato z [12].

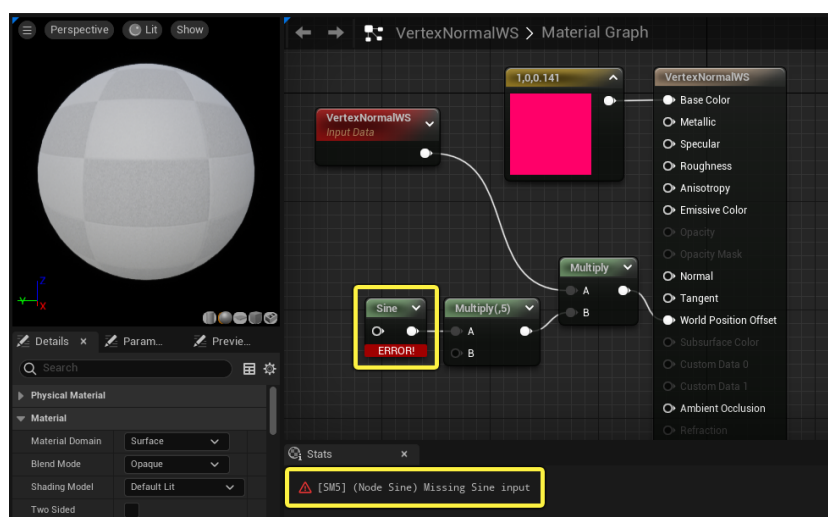
Jednotlivé uzly obsahují datové piny, které jsou barevně odlišeny podle typu přenášené informace. Spojovací body mají kulatý tvar a po připojení drátu se jejich vnitřní část vizuálně zaplní, čímž poskytují zpětnou vazbu o aktivním spojení. Na případné

³ <https://www.unrealengine.com/en-US>.

chyby nebo nesoulad v propojení je uživatel upozorněn prostřednictvím kontextové nápovědy *tooltip*. Za pozornost stojí také mechanismus automatické konverze datových typů při spojování nekompatibilních pinů, pokud je konverze možná, editor ji provede automaticky, což výrazně zvyšuje uživatelský komfort. Podobnou funkcionalitu by bylo vhodné zvážit i v rámci návrhu systému I3T, ale kvůli konverzním operátorům s více vstupy by její implementace mohla být složitější.

V případě editoru materiálů je práce s uzly a dráty v Unreal Engine obdobná jako v systému Blueprint. Za zmínku stojí zejména způsob upozorňování na chyby. Pokud dojde k chybě, tak se ve spodní části příslušného uzlu zobrazí chybová hláška na červeném pozadí, což představuje velmi výraznou vizuální zpětnou vazbu. Podrobnější popis chyby je následně dostupný v dolní liště editoru. Tento způsob signalizace chyb je výrazně názornější než aktuálně implementované řešení v systému I3T, a mohl by proto sloužit jako vhodná inspirace pro jeho další vývoj.

Podobně jako Blender umožňuje seskupování uzlů do bloků pro lepší orientaci v prostoru a spojení označených uzlů do jednoho.



Obrázek 5.5. Vzhled uzlů v editoru materiálů v Unreal Engine, převzato z [13].

5.3 Vytvořené návrhy

Následující sekce podrobně představuje jednotlivé návrhy řešení vybraných problémů, včetně popisu procesu jejich vzniku tam, kde je to relevantní. Návrhy jsou tematicky rozděleny podle oblastí, kterých se týkají. Nejde o úplný výčet všech navržených změn, jelikož některé návrhy vznikly až v pozdějších fázích testování a jiné byly v průběhu vývoje opuštěny. Kompletní přehled všech provedených úprav je uveden v kapitole 7. Řešení, která se týkají výhradně implementačních chyb (např. chybné zobrazení menu mimo obrazovku nebo vykreslování drátů přes ostatní uzly), zde uvedena nejsou. Nejsou uvedeny ani úpravy, jejichž řešení bylo zcela jednoznačné, jako například potřeba zvětšit velikost písma.

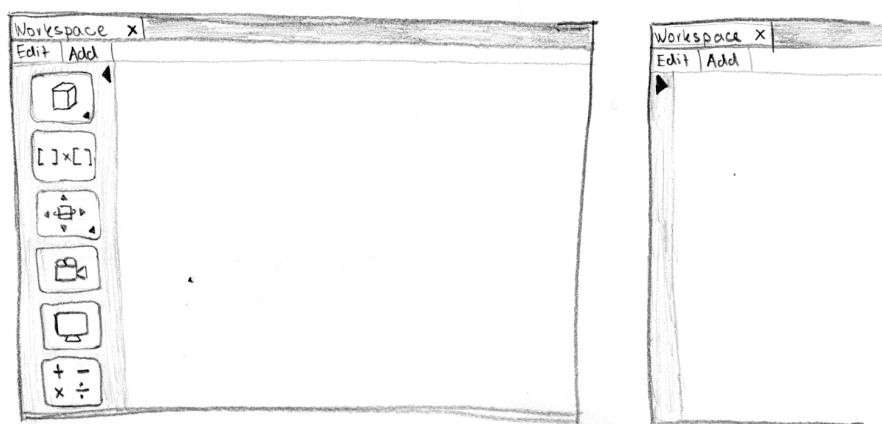
5.3.1 Celý pracovní prostor

Problém s pohybem ve scéně pomocí dotykové podložky (*touchpad*) je způsoben tím, že k posunu je v současné verzi aplikace vyhrazeno prostřední tlačítko myši. Jako řešení se nabízí umožnit pohyb také prostřednictvím pravého tlačítka myši, které je k tomuto

účelu běžně využíváno i v jiných aplikacích. V I3T je však pravé tlačítko již používáno k otevírání kontextového menu pro přidávání objektů. Bude proto nutné otestovat, zda by zavedení alternativního ovládání nevedlo k nežádoucímu chování, například k neúmyslnému otevření kontextového menu při pokusu o posun scény.

5.3.2 Menu

Pro zlepšení naležitelnosti menu jsem navrhla přidat položku „Add“ do horní lišty *workspace*, vedle stávajícího menu Edit. Zároveň jsem považuji za důležité zachovat možnost otevřít toto menu pravým kliknutím do pracovní plochy, jelikož jde o zažitý způsob ovládání, zejména u zkušených uživatelů, a umožňuje rychlejší interakci.



Obrázek 5.6. Návrh ikonového menu ve vysunutě a zasunutě verzi a přidání položky Add do horní lišty *workspace*.

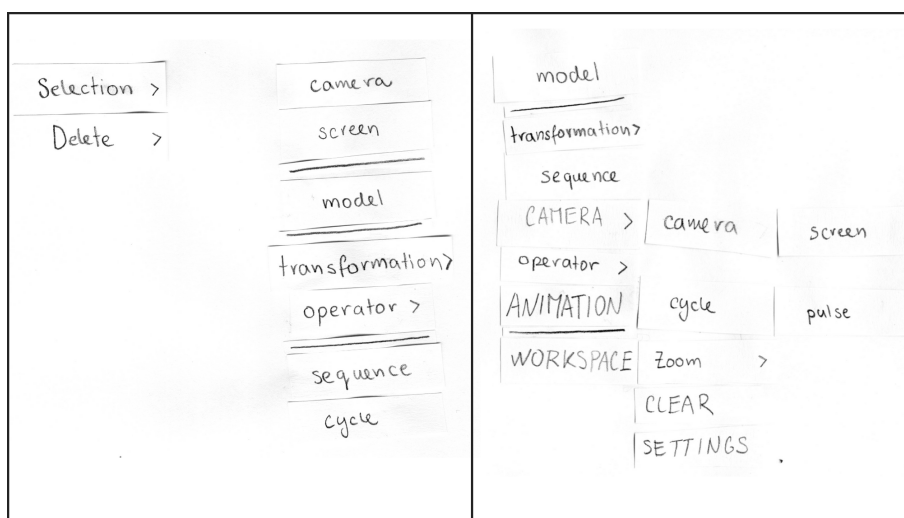
Dále jsem navrhla doplnit program o ikonovou verzi menu (obrázek 5.6), která by sloužila jako vizuálně přehledný a rychle dostupný nástroj k přidávání objektů, především pro začínající uživatele. Rozhodla jsem se pro tento přístup, jelikož ikony mohou výrazně usnadnit orientaci v rozhraní a urychlit práci s často používanými funkcemi. Ikonové menu bych umístila přímo do *workspace* nebo do scény. Aby však nedocházelo k omezování pokročilých uživatelů při jejich běžné práci, navrhla jsem variantu vysouvacího panelu, který je možné dle potřeby snadno skrýt či zobrazit. Tento přístup umožňuje zachovat zavedené pracovní postupy, a zároveň nabízí nápovědu méně zkušeným uživatelům. Klíčovým krokem při návrhu tohoto menu je pečlivý výběr ikon, které by měly jednotlivé funkce srozumitelně a výstižně reprezentovat. Pro výběr ikon využiji testování s uživateli.

Poslední úpravou je změna pořadí položek v přidávacím menu. Hlavním cílem je od sebe oddálit transformace a operátory, aby se snížilo riziko jejich záměny. Součástí vylepšení menu může být také možnost přímého výběru typu modelu již při jeho přidávání do scény, čímž by byli uživatelé upozorněni na existenci více variant modelů. Přidávací menu obsahuje i funkce jako odstranění, označení nebo přiblížení, které však přímo nespojují s přidáváním prvků do pracovního prostoru. Jejich umístění v této nabídce je proto nelogické. Navrhuji tyto funkce přesunout do nabídky Edit v horní liště, která již nyní obsahuje podobné operace, jako je výběr všech uzlů či přiblížení na vybrané uzly.

K návrhu seřazení menu jsem se rozhodla využít uživatelské testování. Toto testování bylo provedeno s využitím metody jménem „card sorting“, což je jedna z metod uživatelského výzkumu a testování, kde účastníci umísťují kartičky do skupin podle určitých kritérií. Účelem této metody je získat povědomí o mentálním modelu uživatelů. [14]

Cílem tohoto testování bylo zjistit jak by uživatelé seskupili jednotlivé části menu k sobě a čím by určili pořadí těchto seskupení. Účastníci testování obdrželi papírky s jednotlivými položkami menu a měli za úkol je seřadit tak, jak jim to přijde nejvíce intuitivní. Také měli k dispozici oddělovače ve formě čáry, kterými mohli více zdůraznit oddělení položek v menu. Zároveň měli možnost libovolné položky vůbec nevyužít, přidělat si vlastní, nebo dostupné položky upravit, dle svých potřeb.

Testování se zúčastnili celkem dva studenti, kteří již s aplikací zacházeli a měli tedy o základních funkcích aplikace přehled. Výsledkem byla dvě nová seřazení menu. Ke každému seřazení dodal účastník testování komentář o tom, jaká učinil rozhodnutí a proč.

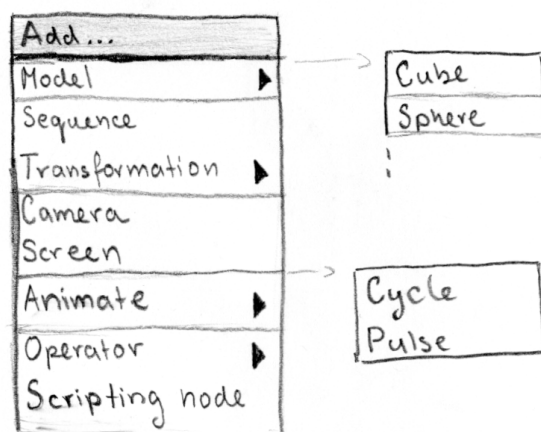


Obrázek 5.7. Seřazení menu z testování.

Oba návrhy se od sebe na první pohled odlišovaly. U prvního návrhu došlo k oddělení možností, které manipulují s *workspace* do samostatného menu, jelikož s přidáváním do scény nesouvisejí. Rozdíly se také objevily v umístění transformací a operátorů. Ve druhém návrhu je větší vzdálenost mezi transformacemi a operátory, což je vhodné pro jejich oddělení, jelikož v současném rozložení dochází k jejich záměně.

Návrhy však také sdílely několik klíčových principů. Oba participantů se shodli na tom, že uspořádání menu by mělo odpovídat logice pořadí kroků při vytváření scény. Společným bodem obou návrhů je také umístění položek kamery a obrazovky *screen* vedle sebe, což dává smysl, protože tyto elementy spolu přímo souvisejí.

Výsledný návrh (obrázek 5.8) jsem vytvořila s ohledem na výše uvedená zjištění. Spodní část původního menu jsem oddělila a přesunula do Edit menu. Zároveň jsem položky rozdělila do pěti tematických kategorií. Položka *model* je umístěna samostatně a přímo umožňuje výběr konkrétního typu modelu. Druhou kategorií tvoří sekvence a transformace, které spolu úzce souvisí a jsou často používány společně. Podobným způsobem jsem seskupila kameru a *screen*, jelikož i tyto funkce bývají zpravidla využívány současně. Na základě návrhů od účastníků testování jsem sloučila položky *cyklus* a *puls* pod jednu kategorii s názvem „Animate“. Poslední skupinu tvoří položky operátory a skript, které slouží k pokročilejšímu ovládání programu. Pořadí jednotlivých kategorií odpovídá logice jejich přidávání do scény. Druhou variantou je úplně odstranit položky transformací z přidávacího menu a umožnit jejich vkládání pouze prostřednictvím sekvence. Tím by se zároveň odstranil problém s jejich aplikováním na *model*, neboť správné propojení transformací s modelem vyžaduje použití sekvence.



Obrázek 5.8. Výsledný návrh přidávacího menu, varianta s transformacemi.

Navržené uspořádání jsem následně otestovala s uživateli s cílem ověřit, zda je nové rozvržení efektivnější a srozumitelnější než původní.

■ 5.3.3 Dráty

Při navrhování způsobu manipulace s dráty bylo klíčové vyřešit jejich odpojování. Prvním návrhem bylo umožnit označení drátů a jejich odstranění pomocí klávesy „Delete“. Alternativní návrh vzešel od kolegy Rakušana a to zcela upustit od možnosti dráty označovat a místo toho umožnit jejich odpojení tažením za koncový vstup. Tento přístup by mohl být pro uživatele přirozenější, jelikož se podobný způsob práce s dráty používá i v jiných aplikacích s uzlovým rozhraním, například v modelovacích softwarech Blender a Maya.

Další navrhované vylepšení se týká usnadnění zapojování drátů. Cílem bylo omezit chybná zapojení tím, že bude vizuálně naznačeno, kam lze drát připojit. Konkrétní podobu tohoto návrhu navrhl kolega Rakušan. Návrhem je, že při potažení za drát dojde k dočasnému zešednutí všech vstupů, do nichž drát nelze připojit.

Užitečnou funkcionalitou, s jejímž návrhem přišel vedoucí práce, je možnost vkládat uzly přímo na dráty. Tento přístup by výrazně zjednodušil práci s programem. V současnosti je nutné při vkládání nového uzlu mezi dvě sekvence ručně odpojit oba konce drátu a následně je znovu propojit přes nový uzel. Efektivnějším řešením by bylo umožnit vložení uzlu přímo na stávající drát, pokud jsou jeho vstup a výstup kompatibilní s typem drátu.

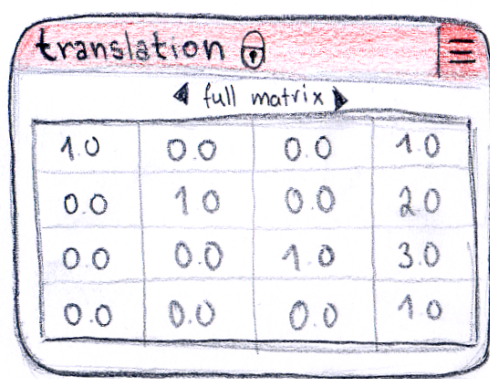
Posledním návrhem na zlepšení práce s dráty je možnost propojit dva uzly jednodušeji, jejich označením a následným stisknutím klávesové zkratky nebo výběrem funkce „propojit“ z menu. Tento způsob by mohl výrazně zrychlit proces napojování uzlů a zároveň snížit chybovost při ručním tahání drátů.

■ 5.3.4 Uzly

Jedním z prvních cílů bylo zpřehlednit a upozornit na existenci kontextových menu uzlů. Stávající řešení dostatečně neupozorňovalo na to, že uzly obsahují kontextová menu, což vedlo k jejich přehlížení. Navrhla jsem proto doplnění tlačítka s ikonou menu do pravého horního rohu každého uzlu, která by na kontextové menu jasně upozorňovala (obrázek 5.9). Zároveň by toto tlačítko umožnilo otevřít menu i levým tlačítkem myši, což je pro řadu uživatelů přirozenější. Pozornost jsem dále věnovala také samotné struk-

tuře menu, aby byla přehledná a snadno pochopitelná. Konkrétní návrhy jsou detailně rozpracovány v následujících částech.

Také jsem řešila tlačítko pro přepínání jednotlivých režimů zobrazení uzlu, které bylo pro uživatele těžké najít. Jako možné řešení mě napadlo přidat přepínač pod horní lištu uzlu, který by obsahoval šipky pro přepínání a název aktuálního režimu (obrázek 5.9). Tento způsob by ale mohl zbytečně zahlcovat samotný uzel a ztěžovat jeho čitelnost. Druhou možností je vymyslet vhodný vizuální indikátor do levého horního rohu místo současných znaků $>$, v , s . Dále by bylo vhodné změnit název položky v kontextovém menu. Aktuální označení „Level of detail“ není příliš srozumitelné. Navrhuji jej nahradit například názvem „Set display mode“, který funkci lépe vystihuje. Pokud má uzel jen dva režimy, stačila by v menu jedna volba, například Rozbalit uzel nebo Sbalit uzel podle toho, v jakém stavu se uzel právě nachází.



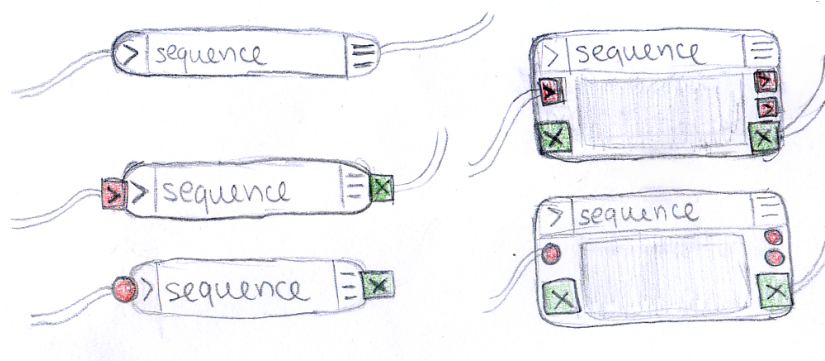
translation 🔒 ☰			
◀ full matrix ▶			
10	0.0	0.0	10
0.0	10	0.0	20
0.0	0.0	10	30
0.0	0.0	0.0	10

Obrázek 5.9. Návrh vzhledu uzlu translace.

Následující návrhy se týkají zlepšení napojování uzlů. Prvním z nich je úprava vertikálního zarovnání vstupních pinů pro maticové násobení, konkrétně jejich posunutí do středu strany uzlu (obrázek 5.9). Takové zarovnání by vizuálně lépe odpovídalo běžnému zápisu násobení matic. Tento návrh vychází z původní verze programu vytvořené Michalem Foltou [1]. Napojení pro násobení by navíc bylo dobré vizuálně odlišit od ostatních napojení, například změnou jejich tvaru. Napadlo mě upravit všechna napojení reprezentující datové typy do kulatého tvaru (obrázek 5.10). V rámci celkové úpravy napojování by také bylo užitečné zvětšit velikost jednotlivých napojení, aby se s nimi lépe pracovalo, a doplnit nápovědu *tooltip*, která by při najetí kurzorem zobrazovala typ napojení.

Dalším bodem, kterým jsem se zabývala, bylo, jak uzly zobrazovat ve sbaleném režimu, zejména z pohledu napojení. Návrhem bylo umožnit manipulaci s napojeními i v tomto režimu, což by usnadnilo rychlou práci. Tento nápad se dá rozdělit na dvě varianty. První možností je, že by byla viditelná všechna napojení, druhou variantou je zobrazení pouze těch napojení, která jsou aktuálně aktivní, což by uživatelům umožnilo je snadno odpojit. Všechny varianty jsou zachyceny na obrázku 5.10.

V případě zamčených hodnot v maticích by bylo vhodné uživatele jasně informovat, že je možné je odblokovat. Prvním nápadem bylo využít nápovědu (*tooltip*) při přejetí přes hodnoty, ale to by nebylo dostatečně přehledné, protože uživatel by musel pro zobrazení nápovědy na hodnoty najet myší. Proto jsem navrhla řešení - umístit do horní části uzlu ikonu zámku, která by signalizovala, že hodnoty jsou uzamčeny, ale



Obrázek 5.10. Návrh vzhledu napojení a sbaleného režimu.

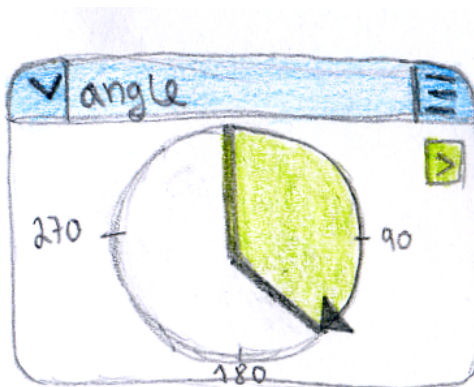
dají se odblokovat, například kliknutím na tuto ikonu. Podobný přístup by mohl být použit i pro indikaci možnosti synergie hodnot.

Drobným vylepšením je implementace upozornění na možnost změny hodnoty v matici pomocí kliknutí myši a posouvání hodnoty doprava nebo doleva. Tato změna by mohla být vizuálně signalizována změnou kurzoru na vodorovné šipky.

Posledním návrhem je možnost vložení uzlu do samostatného okna. Tato funkcionality by mohla být užitečná nejen pro zvětšení zobrazení uzlu obrazovky, ale i pro detailní zobrazení uzlu při výkladu na přednáškách nebo cvičeních.

■ 5.3.5 Úhly

Jedním z vylepšení při zadávání úhlů by mohlo být vytvoření názornějšího rozhraní, které by umožnilo uživateli upravit hodnotu úhlu jiným způsobem než zadáním hodnoty v radiánech. Při představě úhlu si většina uživatelů vybaví prostor mezi dvěma úsečkami, nebo pokud uvažují o úhlu v kontextu rotace, představí si křivku, po které se pohybují body otáčeného modelu. Jedním z možných řešení by bylo reprezentovat úhel pomocí jednotkové kružnice, kde by uživatel mohl úhel vizuálně zvolit, což by zjednodušilo pochopení rotace (obrázek 5.11).

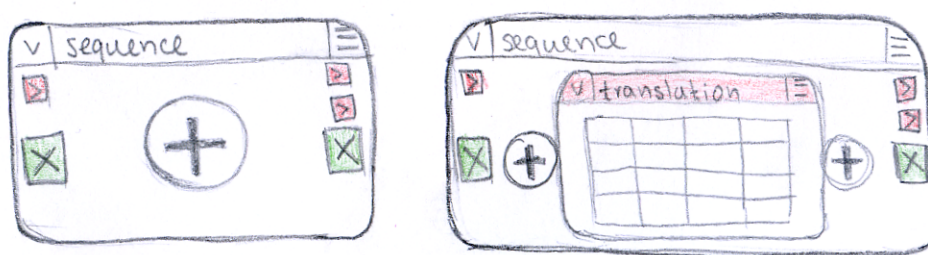


Obrázek 5.11. Návrh rozhraní pro zadávání úhlu.

Dalším nápadem na zlepšení je možnost zadávání a zobrazení úhlu ve formátu $\sin(x)$ nebo $\cos(x)$. Tento formát by mohl být implementován například prostřednictvím převáděcího operátoru, což by zpřehlednilo práci s maticemi rotací a usnadnilo práci uživatelům, kteří s těmito hodnotami manipulují.

5.3.6 Sekvence

Hlavní myšlenkou pro řešení nejasností týkajících se vkládání transformací do sekvencí, bylo navržení vizuální nápovědy k procesu vkládání transformací, umístěné přímo uvnitř sekvence. Jedním z navrhovaných řešení bylo umístit tlačítko s ikonou plus pro přidání transformace do prázdné sekvence (obrázek 5.12). Tento přístup by zároveň umožnil odstranit položku pro přidání transformací z hlavního menu, protože bez existujících sekvencí nemají tyto uzly větší využití. Problematická situace nastává v případě, kdy je již jedna transformace vložena, a je nutné umožnit přidání dalších transformací do jedné sekvence. Možným řešením by mohlo být umístění tlačítek pro přidání transformace na obě strany již vložené transformace, čímž by bylo umožněno přidávat nové transformace do sekvence.



Obrázek 5.12. Návrh sekvence s tlačítkem pro přidávání transformací.

K tomuto řešení navrhl kolega Rakušan alternativní řešení - umístění návodného textu „Drag and drop transformations here“ dovnitř sekvence a zvýraznění pozadí uzlu při interakci, konkrétně při přejetí uzlem transformace nad sekvencí.

Poslední alternativou bylo rozšířit transformace o jejich vlastní napojení, a tím umožnit jejich přímé propojení s modelem. Tato napojení by zanikla při vložení transformace do sekvence, která by tak byla hlavně nástroj na správu více uzlů a spouštění krokování.

5.3.7 Krokování (*tracking*)

Problematické krokování se podrobně věnoval kolega Dan Rakušan ve své diplomové práci [10]. Níže je uvedeno jen několik nápadů a poznatků, které vyplynuly z uživatelského testování. Problém se spouštěním krokování je možné vyřešit odebráním této možnosti z menu u sekvencí, kde krokování spustit nelze, nebo naopak zavedením jednotného způsobu krokování, který bude fungovat pro všechny sekvence. Pro lepší přehlednost by bylo dobré finální model během krokování mírně zprůhlednit, aby byl celý proces krokování viditelný.

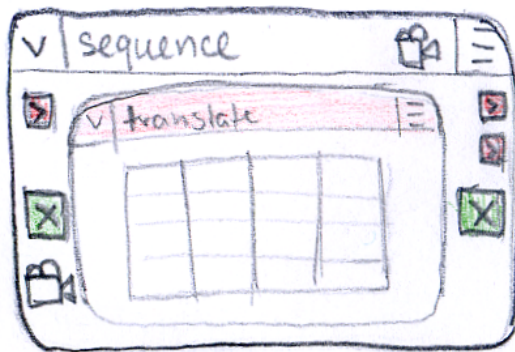
5.3.8 Kamera

Pro uzel kamery bude třeba sjednotit způsob přidávání transformačních matic, konkrétně projekční a pohledové (*projection a view*), s tím, který bude zvolen pro vkládání uzlů do sekvence. Aby byla udržena konzistence interakce v rámci celé aplikace.

Největší nejasnost u uzlu kamery způsobovalo jeho pomyslné napojení na ostatní uzly, které při větším množství prvků ve scéně vedlo k zahlcení a snížení přehlednosti. Prvním návrhem je umožnit uživateli toto napojení volitelně zapínat a vypínat, přičemž ve výchozím stavu by bylo zapnuté, aby uživatelé věděli o jeho existenci. Druhou variantou je toto napojení zcela vypnout a řídit se stejnými pravidly, jaká platí pro napojování

sekvencí a modelů. Tato varianta však narazila na problém - pokud není sekvence nebo model explicitně propojen s kamerou, vyvstává otázka, zda má být pro tuto kameru vůbec viditelný. V rámci scény (*scene view*) jsou tyto objekty stále viditelné, což může vést ke zmatení, proč nejsou zobrazeny i ve výstupu kamery.

Další možností, kterou jsem zvažovala, bylo nahradit vizuální napojení ikonou kamery, která by se zobrazovala v horní liště nebo v levém dolním rohu uzlu a signalizovala, že je daný uzel kamerou zaznamenáván (obrázek 5.13). Tento návrh byl však nakonec zamítnut kvůli nedostatečné názornosti a riziku přehlcení uzlu ikonami.



Obrázek 5.13. Návrh ikon kamery pro uzel.

Poslední změnou je úprava kontextového menu kamery sloučením souvisejících položek do dvou kategorií pro vyšší přehlednost:

- Nastavení barev (výplň a obrys frustum kamery)
- Nastavení viditelnosti jednotlivých komponent kamery ve scéně (osy, model kamery a frustum).

■ 5.3.9 Model

Hlavní změnou pro uzel modelu je úprava a zpřehlednění jeho kontextového menu. Prvním krokem je zjednodušení procesu nastavování barvy modelu. Výběr základní barvy by měl být dostupný přímo z hlavního menu, protože změna barvy slouží převážně k rychlému vizuálnímu odlišení modelů při práci a nevyžaduje přesnou specifikaci odstínu. Z tohoto důvodu by měla být tato akce co nejvíce přímá. Možnost upravení odstínu (*tint*) by byla přesunuta až do další úrovně podmenu. Druhou úpravou pro zvýšení přehlednosti je sloučení všech možností nastavení viditelnosti modelu pod jednu položku, přičemž jednotlivé volby by byly zpřístupněny v samostatném podmenu. Tyto volby nejsou tak často používány a informačně zahlcují menu. Touto změnou by mělo dojít k jeho zpřehlednění.

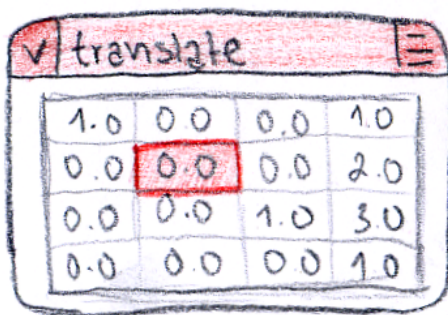
■ 5.3.10 Cyklus

U uzlu cyklu jsem se zamýšlela nad možností úspornějšího zobrazení. Jedním z návrhů je kompaktnější rozložení prvků přiblížením jednotlivých položek k sobě, neboť aktuální rozhraní obsahuje značné množství nevyužitého prostoru. Další úpravou by mohlo být zavedení alternativního režimu zobrazení, který by zobrazoval pouze základní ovládací prvky pro běžné použití. V nejjednodušší podobě by tento režim mohl obsahovat pouze tlačítka pro spuštění a zastavení cyklu, což by bylo vhodné zejména v případech, kdy jsou parametry již nastavené a uživatele zajímá jen samotné ovládání cyklu.

5.3.11 Zpracování chyb (*error handling*)

Stávající nápověda při pokusu o nesprávné zapojení uzlů neposkytuje dostatečné množství informací o chybě. Nápověda by měla konkrétně uvádět typy napojení, která se uživatel pokouší propojit, aby bylo jasné pochopitelné proč k chybě došlo. V některých případech by mohla nápověda poskytovat i informaci, jak problém vyřešit. Například v situaci, kdy se uživatel snaží propojit operátor přímo s modelem, by jej program měl informovat o tom, že je mezi tyto uzly nutné vložit sekvenci.

Upozornění na chybu v matici by mělo být vizuálně výraznější. Jedním z návrhů je zvýraznění chybného pole začervenaním (obrázek 5.14), jelikož červená barva jednoznačně signalizuje chybu. Současná úroveň validace v jádru programu však neumožňuje určit konkrétní pozici chyby v matici, a proto by implementace tohoto řešení vyžadovala rozsáhlejší úpravy, které dosud nebyly realizovány. Dočasným zlepšením by mohlo být zavedení nápovědy (*tooltip*), která by při najetí kurzorem na symbol chyby popisovala konkrétní porušenou podmínku.



Obrázek 5.14. Návrh validace hodnot v matici.

5.3.12 Operátory

Sekce operátorů v přidávacím menu je v této verzi velmi rozsáhlá, což výrazně snižuje její přehlednost. Jedním z možných řešení je sloučit podobné operace, například násobení a sčítání, do jednoho uzlu s možností přepínání konkrétní operace. Taková úprava by však představovala větší zásah do jádra aplikace. Jako jednodušší alternativa se nabízí seskupení příbuzných operací do podmenu, čímž by se snížil počet současně zobrazovaných možností a zlepšila celková orientace v menu.

5.3.13 Další návrhy

Užitečným prvkem pro aplikaci, který zatím neexistuje, by mohly být tzv. bloky. Skupiny sekvencí, případně větší celky navzájem propojených uzlů, by bylo možné sdružovat do bloků, podobně jako je to řešené v programech Blender a Unreal Engine. Tyto bloky by bylo možné jako celek skrýt, uložit a následně znovu načíst, případně hromadně upravovat jejich parametry. Zavedení této funkcionality by výrazně zefektivnilo práci, zejména v případech, kdy se jedná o často opakované části.

Závěrem ještě několik dalších nápadů, které nespádají do žádné z kategorií. Aby program mohl simulovat celou sekvenci transformací aplikovaných na model pro zobrazení na obrazovku, měl by obsahovat ještě uzel pro transformaci záběru (*viewport*). Dalším nápadem je přidat způsob zobrazení složené transformace, která je aplikována na model. Posledním nápadem je možnost přidávat objekty jako jsou model a kamera přímo kliknutím do 3D *scene view*.

Kapitola 6

Iterativní testování

V následující fázi procesu vylepšování uživatelského rozhraní aplikace byly návrhy otestovány s uživateli prostřednictvím prototypů nebo přímo v programu I3T. Na základě získané zpětné vazby byly návrhy dále iterativně upravovány. Tento proces byl několikrát opakován s cílem postupně zvyšovat kvalitu návrhu a tím dojít k optimálnímu řešení a zároveň identifikovat dosud neodhalené nedostatky.

Celkem proběhla tři kola testování. První kolo bylo zaměřeno na ověření návrhů prostřednictvím papírových prototypů. Druhé kolo testování bylo zaměřeno na evaluaci implementovaných návrhů přímo v systému I3T a představovalo hlavní krok při hodnocení dopadu navržených změn na použitelnost uživatelského rozhraní. Třetí kolo se pak věnovalo opravě chyb identifikovaných ve druhém kole a zhodnocení změn provedených dalšími členy týmu, kteří se na vývoji systému I3T podíleli.

6.1 První etapa testování

S ohledem na časové možnosti týmu, který se podílí na vývoji nástroje I3T, jsem pro úvodní testování návrhů zvolila metodu papírových prototypů. Tento přístup umožnil rychlé vizuální ztvárnění navrhovaných úprav. Mezi hlavní výhody této metody patří především časová nenáročnost a jednoduchost, s jakou lze prototypy upravovat, včetně možnosti provádět změny přímo během testování. Dalším přínosem je princip „rychlého selhání“, podle něhož je výhodnější zjistit nedostatky návrhu v rané fázi procesu, než až po jeho plné implementaci. Určitou nevýhodou papírových prototypů je však omezená možnost testování některých typů interakcí. Především situace, kdy je nutné přesně simulovat specifická tlačítka a klávesy nebo konkrétní pohyby kurzoru myši. Návrhy založené na takových interakcích byly proto ponechány k otestování v následující fázi vývoje.

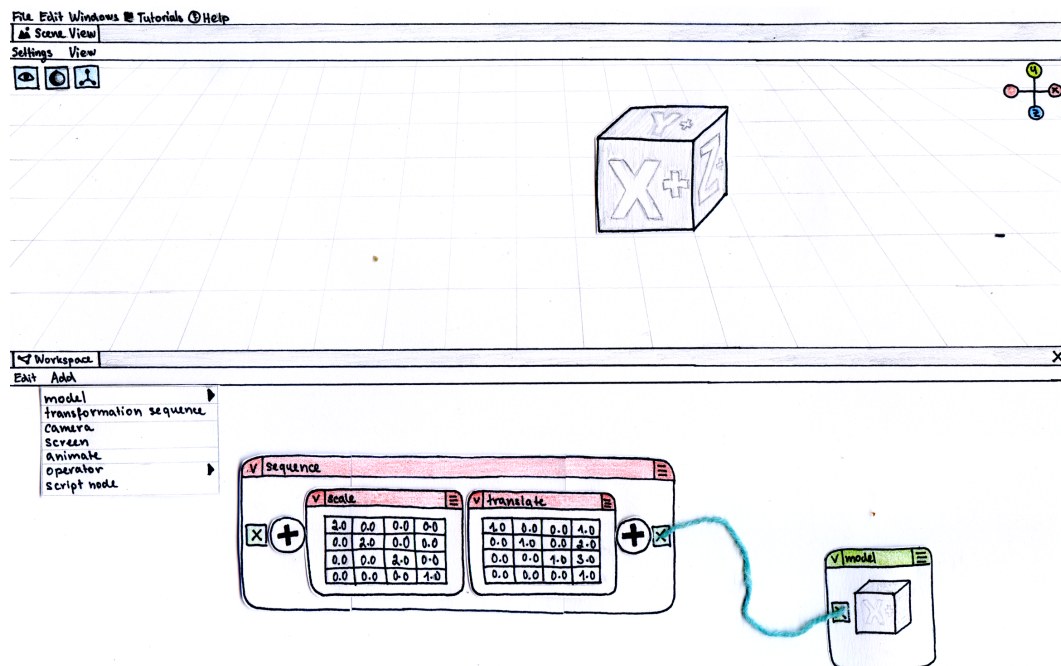
6.1.1 Co bylo testováno

Primárně byly testovány návrhy, jejichž implementace by byla náročná a vyžadovala výraznější zásahy do struktury kódu. Cílem bylo předejít situaci, kdy by byly složité implementovány úpravy, které se následně ukážou jako nevyhovující. Hlavním cílem testování bylo objevit jaký z výše zmíněných způsobů přidávání transformací do sekvence je pro uživatele více přirozený. Pro variantu přidávání do sekvence pomocí tlačítka plus byly testovány dvě možnosti menu, jedno s položkou transformace a jedno bez ní. Dalším testovaným návrhem bylo napojení uzlu obrazovky z levé strany kamery namísto pravé. Zde byly také testovány obě varianty (zleva a zprava) a ty se střídaly během testování. Také jsem testovala zobrazení ikony zámku v horní části uzlu, pokud je matice zamčená a ikony kontextového menu v pravém horním rohu pro jeho snadné nalezení. U modelů byl navíc testován přesun volby typu modelu přímo do hlavního menu a doplnění možnosti importovat vlastní model přímo z této nabídky. Mezi testované změny patřil také způsob odpojování drátů. V rámci papírového prototypu, bylo testování této

funkce mírně omezeno, ale i přesto jsem považovala za důležité pozorovat jakým způsobem mají studenti tendenci dráty odpojovat, aby bylo možné z toho vycházet při implementaci této funkce do programu I3T. Posledním z hlavních návrhů, které byly testovány bylo ikonové menu umístěné na levé straně pracovní plochy *workspace*.

6.1.2 Prototyp

Prototyp (obrázek 6.1) jsem vytvořila z papíru. Nejprve jsem na list A4 nakreslila pracovní plochu a vytvořila několik uzlů reprezentující transformace. Také jsem vytvořila uzel pro sekvenci, který jsem rozdělila v polovině a vytvořila dodatečné prostřední díly, aby bylo možné sekvenci roztáhnout, při vložení více matic. Pro testování nových způsobů vkládání do sekvence jsem ještě připravila tlačítko plus pro vložení transformací a obdélník s textem, který upozorňoval na vkládání transformace. Také jsem vytvořila uzly pro kameru a obrazovku (*screen*) s možností napojení zleva i zprava, aby bylo možné otestovat obě možnosti. Dráty jsem v tomto prototypu reprezentovala pomocí kusů příze, kterými bylo možné uzly propojit. Dále jsem vytvořila ikonu zámku, kterou bylo možné přidat na uzel, pro znázornění zamčené matice. Mimoto jsem nakreslila všechna rozbalovací menu dle návrhů a pro přidávací menu jsem připravila dvě varianty, které si lišily tím, jestli obsahují položku transformace, nebo ne, aby bylo možné otestovat variantu, kdy bude možné vkládat transformace jen ze sekvence.



Obrázek 6.1. Papírový prototyp I3T s variantou přidávání do sekvence pomocí tlačítka plus.

6.1.3 Průběh testování

Účastníky testování jsem vybírala z řad studentů předmětu Programování počítačové grafiky. Tito studenti představují primární cílovou skupinu aplikace I3T, jelikož se teprv seznamují s problematikou transformací v počítačové grafice. Studenti již dříve

pracovali s programem v rámci několika domácích úloh, díky čemuž si osvojili základní funkcionalitu aplikace. Tato znalost se projevila i během testování, kdy někteří účastníci uvedli, že si konkrétní prvky rozhraní pamatují z výukových tutoriálů. Přihlašování na testování probíhalo za pomoci Google kalendáře, kde se studenti mohli zapsat do předem určených časových oken. Každé okno bylo dlouhé 10 minut a mezi nimi byly 5 minutové mezery, které sloužily jako rezerva a prostor pro odpočinek mezi testováním. Testování proběhlo 18.3.2025 v budově školy a zúčastnili se ho celkem 3 studenti. Pro testování jsem připravila jeden úkol, který měli účastníci splnit. Cílem bylo vložit do scény model, změnit jeho barvu, aplikovat transformaci a připojit kameru a obrazovku. Jedná se o velice jednoduchý úkol, ale potřebám testování vyhovoval, jelikož zahrnoval všechny hlavní body testování. Zvlášť jsem se pak ještě účastníků dotazovala na ikonu zámku a také jim odhalila druhou z možností pro přidávání do sekvence a napojení pro obrazovku a kameru a zeptala se, která možnost jim přijde lepší. Možnost, se kterou účastník začínal jsem po každém testování měnila, aby byly zahrnuty všechny varianty. Během testování měli účastníci za úkol nahlas popisovat co dělají. Celé testování probíhalo s papírovými prototypy, což znamenalo, že odezva uživatelského rozhraní nebyla okamžitá. To dávalo prostor pro dotazování se participantů na očekávané chování programu, po provedení akce, což přineslo vhled do mentálního modelu uživatele.

6.1.4 Výsledky

Výsledkem této etapy testování byla spíše rozhodnutí, které verze návrhů mají být implementovány, nebo jestli mají dané návrhy smysl. Naopak identifikace problémů s použitelností aplikace nebyla v této etapě testování hlavním cílem. Výsledkem této fáze testování bylo několik hlavních bodů:

- Pro zdůraznění možnosti přidávání transformací do sekvence byla zvolena metoda návodného textu, „Drag and drop transformations here“. Všichni účastníci testování uvedli, že tento text považují za srozumitelnější a názornější než alternativní variantu ikony se symbolem plus.
- Použití ikony pro kontextové menu přispělo k jeho snadnému nalezení. V rámci testování měli účastníci za úkol změnit barvu modelu, přičemž tato funkce je dostupná právě prostřednictvím kontextového menu. Všichni účastníci nabídku bez obtíží našli, což potvrzuje srozumitelnost nově zvoleného způsobu jejího označení.
- Účastníci testování hodnotili nové rozložení menu jako přehlednější a intuitivnější. Během testování neměl nikdo problém s nalezením požadovaných funkcí, což potvrzuje zlepšení orientace v rozhraní. Zároveň také hodnotili pozitivně možnost výběru modelu již v menu a díky testování objevili možnost nahrání vlastních modelů, o které předtím nevěděli.
- Většině uživatelů připadalo přirozenější odpojovat dráty potažením za jejich konec. Dva ze tří účastníků zvolili tuto metodu, jeden preferoval odpojení kliknutím na drát.
- Testovaný prototyp neobsahoval pomyslné napojení kamery ke všem nezapojeným uzlům v nejvyšší úrovni grafu scény. Účastníci testování si tohoto nedostatku nevšimli, což naznačuje, že vizualizace napojení kamery není pro uživatele nezbytná.
- Řešení ve formě ikony zámku se ukázalo jako nepotřebné. Důvodem může být zvýraznění kontextového menu, díky němuž studenti snadno našli možnost odemknutí. K upozornění na zamknutí stačí pouze zešednutí hodnoty.
- Důležitějším problémem se ukázalo být to, že uživatelé nedokázali snadno najít možnost přepínání režimu zobrazení. Tento nedostatek se projevil jako zásadní a bude nutné jej dále řešit.

- Napojení uzlu obrazovky z levé strany kamery vyvolalo mezi účastníky testování smíšené reakce. Dva účastníci chválili přehlednost, naopak jeden byl zmatený z prohození klasického umístění vstupu a výstupu. Tímto problémem se budu podrobněji zabývat v následujících fázích testování.

Zároveň z těchto výsledků vzešla myšlenka, že k mechanice, která uživatele informuje o možnosti vkládání transformací, je třeba vytvořit i funkcionality, která bude informovat o nutnosti použití sekvence pro aplikování transformace na model. Jedním z nápadů je po delší době, kdy je ve scéně vložen model a uzel transformace, uživatele pomocí nějaké nápovědy informovat, že je třeba do scény umístit i sekvenci. Druhým návrhem je novou scénu vytvořit již s vloženou sekvencí a zapojeným modelem. Což jednoznačně informuje o existenci sekvence.

6.2 Druhá etapa testování

Pro druhou etapu testování jsem se rozhodla změny otestovat přímo v programu I3T, neboť v této fázi již bylo možné je včas implementovat. Při přípravě testování jsem zohlednila poznatky z předchozích kol, na jejichž základě vznikl mimo jiné návrh názornějšího tlačítka pro přepínání režimů zobrazení uzlů. Významná část testování se v této fázi zaměřila také na tvar a umístění napojovacích bodů (*pins*). Součástí příprav bylo i zavedení nového systému pro zaznamenávání uživatelské aktivity, který vytvořil Matvei Korikov [??]. Tento systém umožnil detailně sledovat délku trvání jednotlivých úkonů, identifikovat problematická místa a datově podpořit závěry testování. Kromě toho přidal do programu podporu pro vícejazyčné rozhraní (angličtina, čeština, ruština, ukrajinština), což rovněž přispělo ke zlepšení celkového uživatelského zážitku.

6.2.1 Nové návrhy

Již v úvodní fázi navrhování jsem se zamýšlela nad vzhledem napojení. Hlavní myšlenkou bylo vizuálně oddělit maticové násobení od datových napojení, jelikož reprezentují rozdílné operace a je třeba aby to uživatel na první pohled pochopil. Moje původní návrhy ještě upravil kolega Dan Rakušan, který přišel s návrhem takzvaných socketů. Jedná se o kruhová zapojení, která jsou uvnitř prázdná a po zapojení drátu se vnitřní část vyplní. Původní možnosti a tato nová varianta byly všechny implementovány v I3T (obrázek 6.2) a bylo přidáno menu pro jejich snadné přepínání, aby bylo možné všechny varianty otestovat.



Obrázek 6.2. Možnosti vzhledu napojení.

Také jsem během této fáze vytvořila návrh pro tlačítko přepínání režimů. Hlavní myšlenkou bylo nějakým způsobem reprezentovat to, že má uzel více režimů, než jen sbalení



Obrázek 6.3. Možnosti vzhledu tlačítka pro přepínání režimů.

a rozbalení. Odtud vznikla idea teček které reprezentují počet stavů (obrázek 6.3). Pokud je daný režim zvolen, je jeho tečka vyplněná, jinak je prázdná. Druhým návrhem od kolegy Rakušana bylo nahradit stávající písmena ikonkami. Ikonu šipek pro sbalení a rozbalení menu a ikona pera pro režim nastavení hodnot (obrázek 6.3).

6.2.2 Co bylo testováno

Testování probíhalo s verzí aplikace I3T v2.0.0-rc6 a zaměřovalo se na ověření několika nově implementovaných změn. Jedním z hlavních cílů této fáze testování bylo zjistit, jaký vzhled napojení považují uživatelé za nejpřehlednější, a zda si uvědomují, že napojení, kterým spojují sekvence a modely reprezentuje maticové násobení. Dalším důležitým bodem testování bylo nalezení nejlepší varianty tlačítka pro přepínání režimů zobrazení. Opět bylo ověřováno napojení uzlu obrazovky na uzel kamery, které bylo nově přesunuto na její levou stranu. Mezi další testované funkce patřilo nově implementované odpojování drátů potažením za jejich konec, stejně jako zvětšení aktivní oblasti pro vytvoření a připojení drátu, což mělo zjednodušit manipulaci s propojeními. Pomyslné napojení kamery, které dříve sloužilo jako vizuální indikace propojení, bylo dočasně zcela odstraněno. Dále byla implementována možnost začít novou scénu s předem vloženou sekvencí a modelem, přičemž tuto funkci bylo možné volitelně aktivovat či deaktivovat, aby bylo možné obě možnosti vyzkoušet.

Testování se dále zaměřilo na ověření změn, které byly testovány již v předchozí etapě pomocí prototypů, aby se zjistilo, zda budou stejně dobře fungovat i v samotné aplikaci. Konkrétně šlo o nové rozložení přidávacího a kontextových menu, které jsem implementovala, dále o ikonu kontextového menu umístěnou v pravém horním rohu uzlu a o návodný text uvnitř sekvence.

Druhým cílem testování bylo zjistit, zda nově implementované změny nepřinesly neočekávané problémy a případně objevit nové nedostatky, které v předchozích etapách zůstaly přehlédnuty.

6.2.3 Průběh testování

Ve druhém kole se testování zúčastnili opět studenti předmětu Programování počítačové grafiky. Přihlašování probíhalo stejně jako v předchozím kole prostřednictvím Google Kalendáře. Na rozdíl od minulého testování jsem provedla změnu v motivování studentů k účasti na testování. Vzhledem k nízké účasti v první fázi testování, jsem se dohodla s vedoucím práce na udělení tří bonusových bodů do předmětu PGR za účast. Tato forma motivace se ukázala jako velmi účinná. Všechna připravená místa byla obsazena během několika hodin. Celkem se zúčastnilo deset studentů.

Testování proběhlo ve dvou dnech, 15. a 16. dubna 2025, v prostorách fakulty: první den v místnosti Triangl na Karlově náměstí, druhý den ve studovně v Dejvicích. Průběh testování byl obdobný jako v první fázi. Pro každého účastníka bylo vyhrazeno 15 minut plus 5 minutová rezerva. Studenti obdrželi papírové zadání úkolů a pracovali na

notebooku s novou verzí I3T, vybaveném myší pro pohodlnější ovládání. Každý účastník měl za úkol během testování splnit tři úlohy:

- Vytvořit model a ten otočit o 30° podél osy x a následně posunout o vektor $(1, 2, 3)$. Změnit jeho barvu a změnit počet desetinných míst jednotlivých matic na 2.
- Přidat kameru a transformace pohledu a projekce (view a projection), připojit k ní obrazovku a nastavit hodnoty matic kamery, aby byl předem vložený model dobře viditelný.
- Vložit ještě jeden model a aplikovat na něj inverzi transformací aplikovaných na původní model.

Během plnění úkolů účastníci nahlas popisovali své kroky a myšlenkové postupy. V průběhu testování jsem si vedla poznámky o tom, jak si účastníci vedli, co jim šlo snadno a s čím naopak měli problém. Na závěr testování jsem jim představila i alternativní varianty napojení a přepínání režimu matic a zeptala se, která z nich jim nejlépe vyhovuje. Současně jsem také shromáždila doplňující komentáře a návrhy na zlepšení.

6.2.4 Výsledky

Na základě závěrů testování bylo možné rozhodnout, která řešení mají být zvolena. Pomohlo odhalit několik zatím neobjevených problémů s použitelností, a potvrdilo existenci již dříve identifikovaných problémů, které zatím nebyly vyřešeny. Zde je přehled hlavních poznatků:

- Bylo patrné, že účastníci již měli s programem zkušenosti, což spolu s provedenými úpravami vedlo k výrazně plynulejší práci s aplikací. Všichni účastníci navíc ocenili zvětšenou velikost písma.
- Nalezení menu pro přidání objektů bylo pro uživatele bezproblémové. Značná část účastníků využívala k jeho otevření pravé tlačítko myši, protože tento způsob již znali z předchozí práce s programem. Orientace v nabídce byla celkově dobrá, problém však nastal při hledání specifických položek mezi operátory, konkrétně inverze matice. Jako návrh na zlepšení účastníci uváděli zavedení funkce vyhledávání položek v menu.
- Přítomnost sekvence a modelu při vytvoření nové scény byla hodnocena pozitivně. Během testování se však ukázalo, že by bylo vhodné novou scénu při vytvoření automaticky centrovat a resetovat přiblížení, jelikož se výchozí uzly občas zobrazovaly mimo střed nebo příliš malé.
- Manipulace s dráty stále není zcela bezproblémová. Odpojování stále občas způsobuje potíže a oblast pro připojení by mohla být větší. Také se několik participantů pokoušelo o odpojení drátů kliknutím na koncový vstup, bylo by dobré tuto možnost z původní verze zachovat. Jeden z účastníků navrhl také možnost *snappingu* uvnitř celé plochy uzlu při podržení klávesy Alt, inspirovanou chováním v programu Blender.
- Z testovaných variant pro výběr režimu zobrazení uzlu uživatelé nejčastěji preferovali ikonu tří teček, kterou vnímali jako názornější. V případech, kdy má uzel pouze dva základní režimy, by však upřednostnili použití specifických ikon pro sbalení a rozbalení.
- Z testovaných variant napojení byla nejlépe hodnocena varianta se sockety (kolečky, která se po zapojení vyplní). Většina účastníků však uvedla, že jim konkrétní vizuální podoba napojení nepřijde důležitá. Bylo nicméně patrné, že ti, kteří začínali s variantou socketů nebo kulatých napojení, častěji správně pochopili, že napojení se symbolem $\times$ představuje maticové násobení.

- Vstup pro uzel obrazovky z levé strany uzlu kamery nebyl pro většinu uživatelů na první pohled intuitivní. I přesto nakonec vždy přišli na správný způsob propojení, nicméně preferují zachování tradičního uspořádání vstupu vlevo a výstupu vpravo. Nová varianta napojení však zpřehlednila pracovní plochu, což je pozitivní. Řešením by tedy mohlo být spíše nalezení lepšího způsobu zobrazení vstupu a výstupu pro tyto uzly, než vrácení napojení do původního stavu.
- Nápoje (tooltips) pro chyby v matici ještě nebyly implementovány, ale jejich potřeba byla potvrzena. Uživatelé si často nevšimli symbolu chyby v horní části uzlu, a pokud ano, nechápali proč k chybě došlo. Současný symbol chyby by bylo vhodné nahradit srozumitelnější variantou.
- Nastavování úhlů působilo studentům potíže, tři z nich zadávali hodnoty ve stupních, což ukazuje, že by bylo vhodné buď umožnit zadávání ve stupních, nebo výrazněji označit, že se jedná o radiány. Navíc kvůli omezení na jedno desetinné místo vypadá například $\pi/3$ pouze jako 1, což je matoucí.
- Další častou chybou bylo zadávání velikosti úhlu přímo do matice rotace, místo sin a cos tohoto úhlu. Bylo by zajímavé zvážit možnost automatické konverze, ale je potřeba promyslet způsob, jak to provést srozumitelně, aby nedocházelo ke zmatení uživatelů.
- Menu pro výběr hodnot, které se zobrazuje po kliknutí pravým tlačítkem na hodnotu, se studentům obtížně hledá a působí nepřehledně.
- Účastníci měli problém najít možnost pro resetování matice v kontextovém menu. Bude třeba, buď upravit její umístění, nebo vymyslet srozumitelnější název.
- Někteří uživatelé se snažili přidat matice projekce a lookAt kliknutím přímo dovnitř kamery. Možným řešením k otestování je přidat kontextové menu dostupné pravým klikem dovnitř kamery, které tyto transformace nabídne.
- Již dříve bylo zaznamenáno, že zamčené matice uživatele matou. Během testování se ukázalo, že se problém týká výhradně zamčené matice lookAt. Jako řešení se nabízí zobrazit tento uzel přímo v režimu zadávání hodnot.
- Koncept operátorů není pro studenty srozumitelný. Často nerozumí, proč například operátor inverze zobrazuje matici, kterou ale nelze přímo propojit s modelem. Tento problém by mohla řešit změna operátorů na variantu, která nebude zobrazovat data, případně přidat indikátor, že pro propojení s modelem je nutné využít sekvenci. Alternativně by mohlo pomoci umožnit vložení sekvence přímo z operátoru, nebo přidat tutoriál vysvětlující princip operátorů.
- Funkce výběru (*select*) nefunguje vždy intuitivně. Uzel je vybrán pouze tehdy, pokud je celý uvnitř výběrového rámečku (při tažení z levého horního rohu). To uživatele často mátl, protože očekávali, že bude stačit částečné překrytí.

■ 6.2.5 Porovnání výsledků s úvodním testováním

Na základě druhé etapy testování jsem zpracovala statistické vyhodnocení náročnosti jednotlivých úkonů v aplikaci s cílem porovnat výsledky s úvodním testováním a posoudit míru zlepšení uživatelského rozhraní. V každé z osmi hodnocených kategorií byl každému účastníkovi přiřazen počet bodů v rozsahu 0–3 na základě hodnocení popsaného v sekci 2.4.7. Na základě těchto údajů jsem pro každou kategorii vypočítala průměrnou hodnotu a směrodatnou odchylku. V úvodním kole testování byli hodnoceni tři účastníci, druhé kolo zahrnovalo deset participantů.

Z výsledků vyplývá, že došlo k výraznému zlepšení hodnocení ve dvou klíčových kategoriích, *změna parametrů* a *aplikování transformace na model*, a to v průměru až o 2 body (přibližně o 60%). Zlepšení bylo zaznamenáno také v kategoriích *vyhledávání v*

menu a orientace v pracovním prostoru, kde se průměrné hodnocení zvýšilo přibližně o 1 bod (přibližně o 30%). Ostatní pozorované oblasti vykazovaly buď shodné výsledky, nebo pouze mírné zlepšení. V případě kategorie *vypracování úkolů* je tento výsledek očekávatelný. Naopak u kategorie *zapojování drátů* poukazuje výsledek hodnocení na přetrvávající problémy v této oblasti, kterým je třeba se dále věnovat.

Testovaná oblast	úvodní testování	2. kolo testování
Vyhledávání v menu	2.00 ± 0.58	2.90 ± 0.10
Změna parametrů v menu uzlu	0.66 ± 0.67	2.50 ± 0.22
Nastavení hodnot transformací	2.33 ± 0.33	2.40 ± 0.016
Zapojování drátů	2.00 ± 0.00	2.10 ± 0.18
Aplikování transformace na model	0.66 ± 0.33	2.60 ± 0.16
Vypracování úkolu	2.66 ± 0.33	2.60 ± 0.16
Orientace v pracovním prostoru	1.66 ± 0.33	2.70 ± 0.15
Porozumění aplikaci	2.33 ± 0.33	2.60 ± 0.16

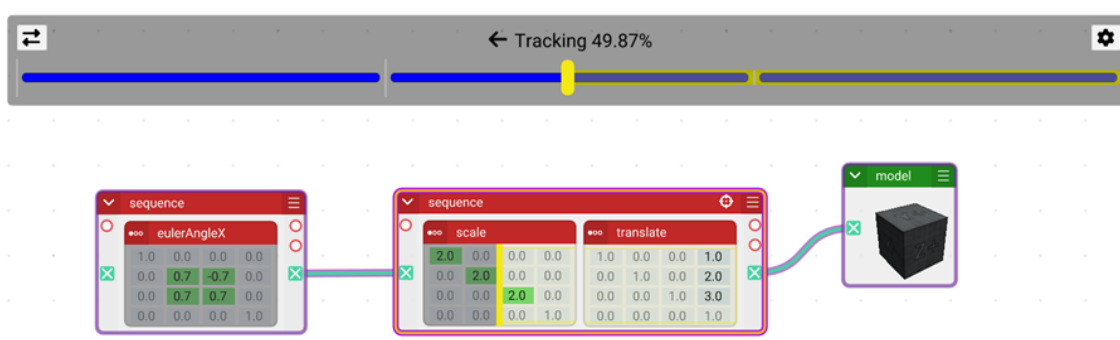
Tabulka 6.1. Tabulka porovnávající výsledky úvodního testování a druhého kola iterativního testování (průměr $\pm$ směrodatná odchylka).

6.3 Třetí etapa testování

Poslední etapa testování byla z velké části zaměřena na nově implementovanou funkci krokování (*tracking*), kterou do programu přidal kolega Dan Rakušan jako součást své diplomové práce [10]. Kromě toho se soustředila také na otestování drobných úprav uživatelského rozhraní, které zatím nebyly implementovány v předchozích verzích, a na doladění některých detailů, které vyplynuly z předchozích kol testování.

6.3.1 Co bylo testováno

Mezi nové úpravy, které byly implementovány do programu a podrobeny testování, patří umožnit pohyb po pracovní ploše pomocí pravého tlačítka myši a přidání nápověd (*tooltips*) k napojením, které informují o typu napojení. Upraveny byly také nápovědy, které se zobrazují při nesprávném napojení.



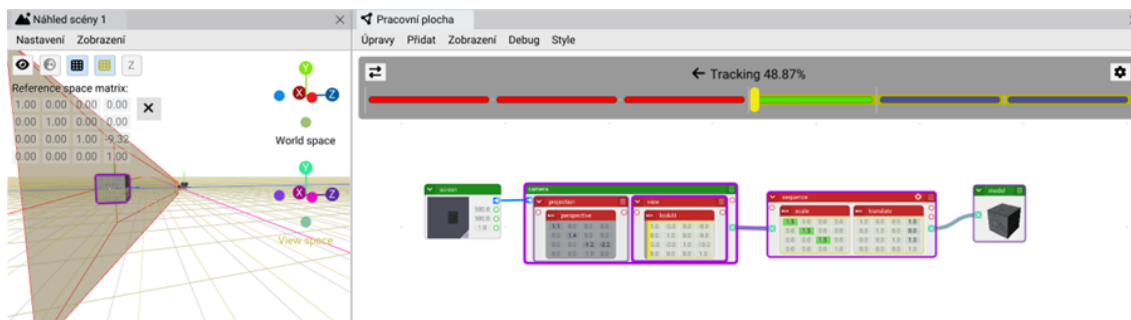
Obrázek 6.4. Nový systém krokování.

Jak už bylo zmíněno, hlavní součástí testování byla nová funkce krokování. Funkce se spouští z kontextového menu sekvence. Pro její ovládání vytvořil kolega Rakušan ná-zorné rozhraní umístěné v horní části pracovní plochy (obrázek 6.4). Toto rozhraní se

skládá z několika prvků: posuvníku (*slider*), který umožňuje pohybovat se skrze jednotlivé kroky aplikovaných transformací, pohyb je také možný pomocí šipek na klávesnici. Dále rozhraní obsahuje tlačítko pro přepínání směru aplikace transformací (zprava - z pohledu modelu, nebo zleva - z pohledu soustavy souřadnic). V pravém horním rohu se nachází menu s pokročilejším nastavením, například pro úpravu rychlosti přehrávání nebo pro úplné vypnutí krokování. Kromě menu je možné krokování ukončit také stisknutím klávesy Esc. Krokování umožňuje pohyb jen směrem doleva od zvolené sekvence a vše co je umístěno nalevo je bráno jako jeden objekt, na který jsou aplikovány transformace, zvolená sekvence je ve *workspace* označena ikonou.

Hlavním cílem této části testování bylo zjistit, jak se bude studentům s rozhraním zacházet a zda správně pochopí, co se jim vizualizace jednotlivých kroků transformací snaží přiblížit.

Rozšířenou verzí klasického krokování je krokování pro kameru (obrázek 6.5), které lze spustit z kontextového menu kamery. Krokování kamery se skládá ze dvou hlavních částí. První z nich je transformace lookAt, která umístí kameru do počátku souřadnicového systému a vytvoří tzv. view space. Druhou částí je transformace projekce, která může být dále rozdělena na různé matice, podle typu dekompozice. Tato transformace transformuje pohled kamery do finálních normálových souřadnic zařízení (*normal device coordinates*). Obě části jsou v rozhraní barevně odlišeny pro snadnější orientaci. Během krokování kamery dochází také k vizuálním změnám ve scéně (*scene view*). Například se objeví nová mřížka pro *view space* a modifikuje se kompas v pravém horním rohu, aby odrazil změny způsobené transformacemi.



Obrázek 6.5. Krokování kamery.

Dále je možné obě části krokování spojit a zobrazit kompletní řetězec transformací. Pro sledování tohoto řetězce musí být požadovaná sekvence přímo napojena na kameru pomocí napojení maticového násobení.

Poslední část testování se zaměřila na výběr ikon, kde bylo cílem zjistit, jak uživatelé vnímají různé předpřipravené ikony. Test měl pomoci určit, které ikony jsou pro uživatele nejsrozumitelnější a nejlépe vyjadřují konkrétní funkci menu, kterou mají zastupovat a bylo je možné využít do připravovaného ikonového menu.

6.3.2 Průběh testování

Výběr účastníků testování probíhal stejně jako v předchozích etapách. Testování se konalo 7.5.2025 ve studovně v Dejvicích a zúčastnilo se ho celkem 5 studentů. K testování se připojil také Matvei Korikov, který v rámci své bakalářské práce [15] testoval rozložení oken a optimalizační nástroje programu I3T. Testovací sezení bylo rozděleno na tři části.

První část vedl kolega Korikov, který měl pro studenty připravených sedm úkolů týkajících se nastavení aplikace. Cílem bylo zjistit, zda jsou tato nastavení intuitivní. Tato část se jen okrajově týkala mé práce, a proto se jí nebudu věnovat.

Druhá část testování se zaměřila na krokování. Studenti pracovali se dvěma předpřipravenými scénami. První scéna obsahovala několik napojených sekvencí a modelů. Úkolem účastníků bylo přidat sekvenci s dvěma transformacemi, zapojit ji do řetězce a následně spustit krokování, nejprve z jedné a poté z druhé strany. Během této části jsem účastníky nechala popisovat jak vnímají to, co se při krokování děje. Druhá scéna obsahovala navíc uzly kamery. Studenti měli opět za úkol spustit krokování, sledovat, jak se mění během postupného aplikování transformací scény, a popisovat své myšlenky během tohoto procesu.

Poslední částí bylo testování ikon pro přidávací menu. Měla jsem připravené vytištěné ikony, které jsem jednotlivě ukazovala účastníkům a ptala se jich, co si pod každou ikonou představují. Jejich odpovědi jsem si zapisovala, abych mohla jednotlivé odpovědi mezi sebou porovnat.

6.3.3 Výsledky

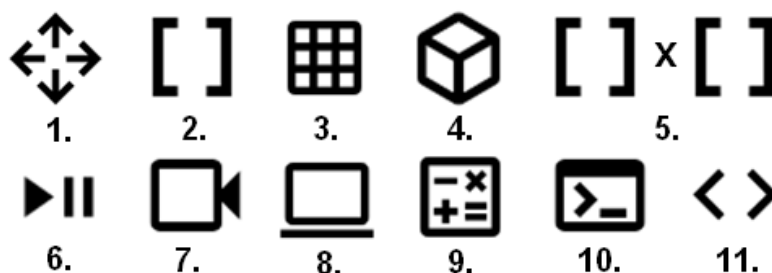
Nově přidané funkce krokování byly celkově velmi pozitivně hodnoceny a používání programu působilo mnohem plynuleji. Níže je uveden přehled klíčových poznatků.

- Poměrně často se stává, že místo očekávaného drag-and-drop pohybu při vytahování uzlů ze sekvencí a tahání za dráty dojde pouze k označení uzlu a žádná akce se neprovede.
- Někteří uživatelé měli potíže s nalezením funkce krokování, kterou zprvu hledali v jiných částech rozhraní a různých menu. Funkci se jim podařilo nalézt až po vysvětlení, že funkce souvisí s uzlem sekvence.
- Použití šoupátka (*slider*) byl pro uživatele intuitivní, bez obtíží s ním interagovali a správně pochopili, že jednotlivé části šoupátka reprezentují sekvence a matice.
- Uživatelům nebylo zcela jasné, proč se do krokování nezapojují modely, které byly po cestě dodatečně připojeny.
- Nad tím, kde krokování začíná, se uživatelé při práci nijak hlouběji nezamýšleli. Při zpětném dotazu, proč se některé části účastní krokování a jiné ne, však nebyli schopni sami situaci vysvětlit. Po vysvětlení však princip pochopili, a proto by mělo být dostačující předat tuto informaci prostřednictvím tutoriálu.
- Většina studentů nenalezla nebo nepochopila funkci tlačítka pro přepínání směru krokování zleva a zprava. Často se domnívali, že tlačítko slouží například k přesunu na začátek či konec, nebo k postupnému krokování. Tato funkce bude vyžadovat výraznější vizuální odlišení nebo doplnění nápovědy.
- Uživatelé by ocenili možnost automatického přehrávání krokování, která by mohla zefektivnit a zjednodušit sledování průběhu sekvence bez nutnosti ručního ovládání.
- Ikony označující počáteční sekvenci si většina uživatelů nevšimla a její význam pochopili až po upozornění.
- Pro některé studenty bylo obtížné představit si průběh krokování transformací zleva, což vedlo k určité míře zmatení. Řešením by mohlo být vizuální znázornění samotného postupu pomocí sledování šipek reprezentujících souřadnicový systém, zatímco samotný model by byl vykreslen až na konci procesu.
- Pochopení konceptu view space, změny souřadnicového systému a nové mřížky nedělalo účastníkům problému.
- Pár účastníkům nebylo zcela jasné, proč se v červené sekci projekční transformace nachází tři části. Po vysvětlení však tuto strukturu pochopili. Bylo by užitečné při-

dat popisky jednotlivých částí, i když celkově účastníci působili dojmem, že princip transformace je pro ně jasný.

- Červená a zelená barva sekcí v horní liště krokování působily na dva participanty dojmem, že červená označuje chybu a zelená správnou funkci.
- Nutnost napojení kamery na část grafu scény, který se účastní krokování, byla pro některé uživatele matoucí. Po upozornění se jim podařilo kameru správně napojit, nicméně byli zmateni z toho, že kamera stále zobrazuje i objekty, které nejsou zapojené.

Výsledkem testování ikon bylo přiřazení popisů, které vyjadřovaly představy uživatelů o jejich významu. Některé ikony byly uživatelům jasné a všichni testovaní se na významu shodli, zatímco jiné pro uživatele představovaly něco úplně jiného, než jsem původně zamýšlela. Následuje popis k jednotlivým ikonám (obrázek 6.6).



Obrázek 6.6. Testované ikony.

1. ikona byla jednou se zamýšlených pro transformace, ale uživatelé si představovali spíše posun.
2. ikona měla také reprezentovat transformace, přičemž uživatelé si ji spojovali především s maticemi, což by mohlo být dostatečné spojení.
3. ikona byla poslední variantou pro transformace. Uživatelům připomínala spíše mřížku.
4. ikona podle všech účastníků testování reprezentovala model nebo objekt.
5. ikona připomínala většině participantů násobení matic, což by mohlo být pro sekvenci dostatečně výstižné.
6. ikona měla reprezentovat cyklus a puls. Většina uživatelů si ji spojovala se spuštěním nebo animací, což jsou významy, které jsou s původním záměrem dostatečně blízké.
7. ikona kamery byla všem jasná.
8. ikona měla reprezentovat obrazovku (*screen*), ale připomínala uživatelům spíše celou obrazovku počítače.
9. ikona reprezentovala operátory, s čímž participanti souhlasili.
10. ikona měla představovat skriptovací uzal, ale všichni účastníci si vybavili terminál.
11. ikona připomínala uživatelům spíše krokování než skript, který měla reprezentovat.

Dosud nebyl nalezen kompletní seznam ikon pro celé menu. Práce na výběru a přiřazení správných a výstižných ikon bude pokračovat v dalších fázích vývoje nad rámec bakalářské práce.

Toto testování představovalo poslední fázi testování v rámci této práce, nicméně aplikace bude i nadále vyžadovat vylepšení. Je důležité pokračovat v analýze a řešení identifikovaných nedostatků v dalších fázích vývoje, aby bylo dosaženo co nejlepšího uživatelského zážitku.

Kapitola 7

Přehled změn v programu

Tato kapitola obsahuje souhrnný přehled úprav, které byly, na základě mých návrhů, provedeny s cílem zlepšit použitelnost programu. Pro každou změnu je uveden původní stav, popis navrženého a implementovaného řešení a zhodnocení jeho úspěšnosti.

- **Pohyb ve workspace pro notebook.** Pohyb po pracovní ploše byl původně možný pouze použitím prostředního tlačítka myši, což výrazně omezovalo uživatele využívající výhradně notebook. Tento problém byl vyřešen přidáním možnosti pohybu i pravým tlačítkem, což výrazně zjednodušilo používání aplikace i na noteboocích.
- **Zavedení škálování rozhraní.** Původní verze aplikace měla problém s malou velikostí písma, což ztěžovalo čitelnost a orientaci v pracovním prostoru. Tento problém byl vyřešen implementací DPI škálování, které si uživatel může přizpůsobit v nastavení. Výsledkem je výrazně lepší vzhled rozhraní a efektivnější používání aplikace.
- **Přesun přidávacího menu do horní lišty.** Začínající uživatelé nemohli najít přidávací menu, a tím pádem nevěděli, jak začít. Řešením bylo umístit tlačítko pro otevření menu trvale do horní lišty *workspace*. Toto řešení umožnilo novým uživatelům rychle najít přidávací menu, ale po delší práci s programem všichni uživatelé otevírají menu původním způsobem - pravým kliknutím do pracovní plochy.
- **Úprava rozložení přidávacího menu.** Původní rozložení přidávacího menu bylo nepřehledné a vyhledávání konkrétní položky trvalo uživatelům dlouho. Bylo navrženo nové uspořádání (obrázek 7.1), které reflektuje pořadí používání položek a jejich vzájemné souvislosti. Do menu byl také přidán výběr typu modelu a položka „Sequence“ byla přejmenována na „Matrix Sequence“. Kromě toho byla odstraněna chyba, kdy se menu otevíralo částečně mimo obrazovku. Tyto úpravy zlepšily přehlednost menu a efektivitu práce s programem.



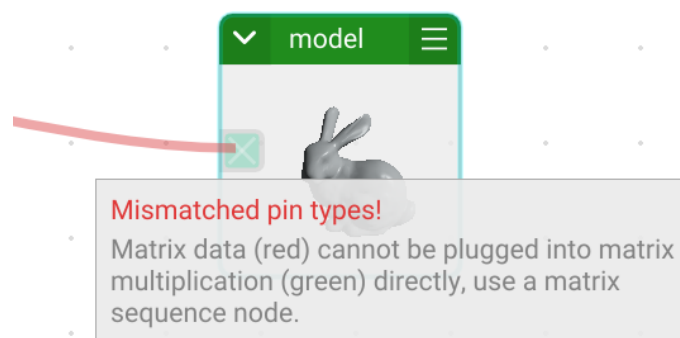
Obrázek 7.1. Finální vzhled Přidávacího menu.

- **Přidání ikony kontextového menu do uzlů.** Uzly nabízejí mnoho možností pro způsobení, ale uživatelé na ně nebyli dostatečně upozorněni, a nemohli tyto funkce snadno najít. Řešením bylo umístit ikonu menu do pravého horního rohu uzlu (obrázek 7.2). Tato změna vedla k podobnému výsledku jako přidání přidávacího menu do horní lišty: ikona slouží jako indikace existence menu, ale po delší době používání aplikace většina uživatelů přejde na používání pravého kliknutí na uzel.
- **Vylepšení rozložení kontextových menu.** Druhým problémem kontextových menu byla obtížná orientace v jejich položkách. Proto došlo k přerovnání a sjednocení několika položek, což vedlo k vytvoření kratších a přehlednějších menu. U některých uzlů byly provedeny větší úpravy, například změna způsobu, jakým se mění barva modelu, což bylo dříve zbytečně složité. Celkově byly změny v uspořádání položek pozitivně hodnoceny a uživatelům se dařilo jednotlivé funkce v novém uspořádání efektivně vyhledávat.
- **Přidání ikony pro změnu režimu do uzlů.** Další funkcionalitou, kterou bylo pro uživatele těžké najít, bylo tlačítko pro přepínání režimů zobrazení uzlů, umístěné v levém horním rohu uzlu (obrázek 7.2). Původní označení tlačítka znaky v , s a $>$ nebylo dostatečně intuitivní. V rámci úprav byla tato označení nahrazena ikonami pro sbalení a rozbalení, pokud uzel nabízí pouze tyto dvě funkce, a ikonou tří teček, pokud má více možností. Tato změna byla uživateli pozitivně přijata, přičemž mnoho z nich díky nové ikoně objevilo třetí typ zobrazení transformací.



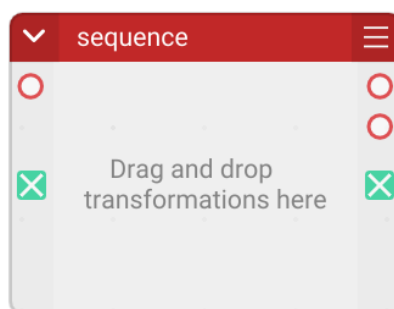
Obrázek 7.2. Finální vzhled uzlu. Nová tlačítka pro kontextové menu a přepínání režimů.

- **Zavedení upozornění na způsob změny hodnot v matici.** Během testování bylo zjištěno, že uživatelé nevědí, že mohou hodnoty v uzlech měnit nejen přímým vepisováním, ale i podržením tlačítka myši nad hodnotou a jejím tažením do stran. Tento problém byl vyřešen změnou kurzoru na horizontální šipky, pokud uživatel najede myší na pole s hodnotou.
- **Úprava nápovědy u zapojení uzlů.** V programu docházelo k častým chybným zapojením uzlů a chybové hlášky byly příliš obecné. Tento problém byl vyřešen přidáním popisků (*tooltips*) k jednotlivým napojením a úpravou stávajících chybových hlášek, aby byly podrobnější a obsahovaly konkrétní návod, jak chybu opravit (obrázek 7.3). Implementace není zatím dokončena pro všechna zapojení, ale dosavadní změny výrazně zpřehlednily proces zapojování drátů.
- **Zešednutí napojení při zapojování drátů.** Další změnou, která řeší problém nesprávných a nejasných zapojení, je zešednutí všech napojení, do kterých nelze ze zvoleného napojení drát zapojit. Tato úprava ještě výrazněji upozorňuje uživatele na správná místa pro zapojení drátu. Zavedení této změny přispělo k lepší orientaci při zapojování drátů.



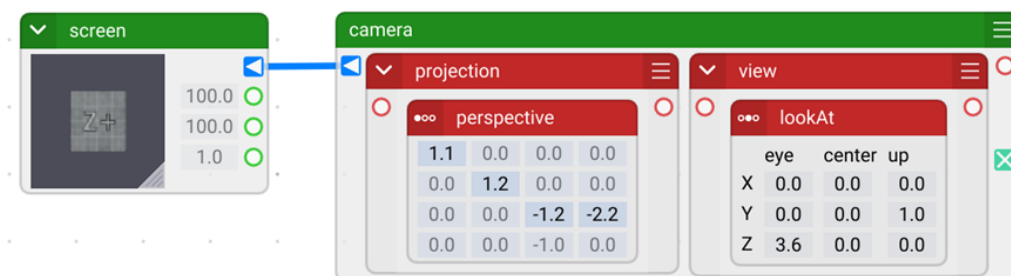
Obrázek 7.3. Nová nápověda pro špatné zapojení uzlů datového typu matice a maticového násobení.

- **Úprava vzhledu napojení.** Uživatelé měli problém s rozpoznáváním napojení uzlů, i když byla rozlišená barvou. Dalším problémem byla jejich velikost, což ztěžovalo přesnou interakci. Tento problém byl vyřešen změnou vzhledu napojení: datová napojení jsou nyní kulatá, zatímco ostatní napojení jsou čtvercová a pro lepší manipulaci byla napojení mírně zvětšena.
- **Změna rozložení napojení.** S předchozím problémem souvisí i skutečnost, že uživatelé nerozpoznali napojení pro maticové násobení. Aby bylo jasně zřejmé, že se jedná o reprezentaci maticového násobení, byla napojení přesunuta do středu matice, čímž se více vizuálně přiblížila konceptu násobení mezi maticemi (obrázek 7.4). Tato změna ve většině případů pomohla k rozpoznání maticového násobení, ale někteří uživatelé stále vnímají tento typ napojení pouze jako další datový typ.
- **Revize manipulace s dráty.** V předchozí verzi aplikace nebyla manipulace s dráty ideální. Jejich odstranění bylo možné pouze označením a výběrem možnosti odstranění z nabídky, nebo kliknutím na koncový vstup. Nové řešení se inspirovalo programem Blender, kde se dráty odpojují potažením. Také byla odstraněna možnost zvýraznění drátů, jelikož nejsou samostatnými objekty. Chyba s překrýváním byla vyřešena a díky nové průhlednosti uzlů je stále přehledně vidět, kudy drát vede. Manipulace s dráty ještě není dokonalá, ale došlo k výraznému zlepšení.
- **Vylepšení vkládání transformací do sekvence.** Dalším zásadním problémem, který vzešel z testování, bylo zmatení uživatelů z toho, že k napojení transformace na model musí využít sekvenci. Tento problém byl vyřešen dvěma změnami. Z nichž první bylo upozornění uživatele na možnost vložení transformace do sekvence pomocí textu uvnitř sekvence (obrázek 7.4).



Obrázek 7.4. Nový vzhled prázdné sekvence s návodným textem uvnitř.

- **Zavedení předvyplněné scény** Druhou změnou, která řeší problém s vkládáním transformací do sekvence, bylo zajistit, že uživatelé budou vědět, že sekvence existuje a že ji mohou napojit na model. Toho bylo dosaženo tím, že nová scéna se otevře již se sekvencí a napojeným modelem. Tento přístup úspěšně vyřešil problém, ale pro některé uživatele je nepohodlné mazání těchto uzlů, pokud skutečně chtějí prázdnou scénu a nechtějí použít model.
- **Vylepšení krokování transformací.** Program umožňuje sledování postupného aplikování transformací na model, ale jeho původní implementace nebyla ideální. Byla tedy vylepšena přidáním rozhraní pro posouvání (*slider*), přidáním menu pro ovládání krokování a zprůhledněním finálního modelu. Tato vylepšení výrazně zlepšila vizualizaci aplikování transformací a zvýšila názornost programu.
- **Vytvoření krokování transformací kamery.** Úplně novým prvkem, který neřeší přímo problém použitelnosti, ale spíše doplňuje funkce programu, je sledování kamerových transformací view a projection. Tato funkce poskytuje lepší pochopení toho, jak tyto transformace fungují v praxi a její přidání mělo pozitivní ohlasy.
- **Revize implicitního napojení kamery.** Dalším prvkem, který byl řešen, bylo implicitní napojení kamery na každý kořen transformací ve scéně, což bylo pro uživatele nejasné a při velkém množství transformací zahlcovalo scénu. Během testování došlo k jeho odstranění a přidání možnosti volitelného plného zapojení pro krokování kamery. Studenti napojení kamery nepostrádali, ale bez něj je vizuální spojení celého řetězce transformací narušeno. Tato změna vyžaduje doladění.
- **Změna strany napojení pro obrazovku.** Problémem bylo, že obrazovka byla napojena z pravé strany kamery, což působilo nepřehledně, protože transformace byly na kameru napojeny ze stejné strany. Řešením bylo přesunutí napojení pro obrazovku na druhou stranu kamery (obrázek 7.5). Tato změna měla smíšené ohlasy, část uživatelů ocenila přehlednost, zatímco jiní kritizovali neintuitivní prohození vstupu a výstupu.



Obrázek 7.5. Finální vzhled uzlů kamery a obrazovky.

- **Úprava vkládání lookAt matice.** Výchozím zobrazením transformace lookAt byl režim, kde jsou všechny hodnoty zamčené. Pro upravení hodnot je třeba přepnout na režim s možností úpravy hodnot pomocí vektorů. To bylo pro uživatele matoucí, jelikož nevěděli, zda mohou s transformací pracovat. Řešením bylo vkládat lookAt transformaci přímo v režimu nastavování hodnot.

Kapitola 8

Diskuze

Do projektu jsem vstoupila až ve fázi, kdy byla aplikace již téměř dokončena, což zásadně ovlivnilo možnosti úprav z hlediska použitelnosti. Odborná literatura zabývající se UX a designem opakovaně zdůrazňuje, že otázce použitelnosti je třeba se věnovat již od počátku vývoje, což však v tomto případě nebylo možné [6, 8]. Zároveň nebylo reálné provádět výraznější zásahy do struktury programu, jelikož mnohé návrhy by narušily již zavedený systém nebo zasáhly do funkcionalit aplikace.

V průběhu jednotlivých iterací se postupně zlepšovala nejen samotná aplikace, ale i můj přístup k testování. Každé testovací kolo přineslo příležitost reflektovat na zvolenou metodiku a dále ji rozvíjet. Ukázalo se, že je vhodné věnovat větší pozornost motivaci účastníků. Zatímco při prvním testování byl počet studentů nedostatečný, druhého testování se účastnilo deset studentů, což se ukázalo jako nadbytečné. Toto zjištění dobře potvrzuje teoretické poznatky, podle nichž postačuje i menší vzorek přibližně pěti uživatelů [7]. Také se vyvíjel samotný způsob vedení testování, od pouhého pozorování se posunul k hlubšímu zkoumání toho, co uživatelé dělají a jak nad aplikací uvažují.

V rámci testování jsem se cíleně zaměřila na začínající uživatele, protože je klíčové zajistit, aby aplikace byla přístupná a srozumitelná již při prvním kontaktu. Tato strategie měla za cíl neodradit nové uživatele od využívání nástroje a usnadnit jim jeho osvojování. Do budoucna by bylo zajímavé zaměřit se také na potřeby a zkušenosti pokročilejších uživatelů, jejichž pohled může přinést další cenné podněty pro zlepšení.

Je rovněž důležité upozornit na možné zkreslení výsledků (*tzv. biases*), ke kterému mohlo během testování docházet. V některých případech byli účastníci motivováni k účasti nabídkou bodového ohodnocení do předmětu, což mohlo vést k menší snaze při testování. Dalším možným zdrojem zkreslení byla rozdílná míra seznámení s aplikací, zatímco účastníci úvodního testování neměli k dispozici žádné tutoriály, studenti v pozdějších etapách již několika tutoriály prošli. Tato skutečnost mohla ovlivnit vnímanou míru zlepšení aplikace, která tak mohla působit větší, než ve skutečnosti byla.

Zajímavým aspektem bylo porovnání výsledků mého testování s poznatky studentů, kteří se otázkou použitelnosti této aplikace zabývali v předchozích letech. Ve většině případů docházelo k výrazné shodě v závěrech, což potvrzuje konzistenci identifikovaných problémů. Již v původní verzi aplikace, na níž pracoval Michal Folta [1], se při testování objevila myšlenka přidávání transformací přímo do sekvence, což ukazuje, že se některé poznatky objevují opakovaně napříč uživatelskými testy. Naopak zajímavým kontrastem byl výsledek práce Víta Zadiny [9], který doporučoval zahájit práci s aplikací s prázdnou scénou. Můj výzkum naopak ukázal, že předpřipravená scéna je pro začínající uživatele vhodnější. Důvodem této neshody je pravděpodobně rozdílný typ testovaných osob, v Zadinově práci výsledek vycházel z testování s pokročilejšími uživateli, zatímco moje závěry reflektují potřeby uživatelů, kteří s aplikací teprve začínají.

Významným prvkem celé práce byla týmová spolupráce. Na projektu I3T jsem pracovala společně s dalšími třemi studenty Danem Rakušanem, Matveiem Korikovem a Karolínou Zapletalovou, což umožňovalo vzájemnou spolupráci. V praxi to znamenalo například to, že pokud potřeboval některý z mých kolegů otestovat novou funkci, zahr-

nula jsem ji do svého testování a jen předala zpětnou vazbu. Zároveň jsem mohla část návrhů na zlepšení použitelnosti delegovat k implementaci ostatním členům týmu, což významně zvýšilo efektivitu celé práce a umožnilo rychlejší reakce na zjištěné problémy.

8.1 Návrhy do budoucna

Práce na zajištění použitelnosti aplikace je dlouhodobý proces, který přesahuje rámec této bakalářské práce. Řešení použitelnosti systému I3T nekončí mým zapojením, a proto by bylo vhodné, aby se této oblasti i nadále věnoval někdo další. V následující sekci proto uvádím seznam návrhů, které byly v rámci testování validovány, avšak dosud nebyly implementovány, dále návrhy, jejichž použitelnost by bylo vhodné dále ověřit, a v neposlední řadě i přehled oblastí, které vyžadují hlubší pozornost. Tyto podněty mohou sloužit jako inspirace a výchozí body pro další bakalářské nebo diplomové práce. Tato část práce je rovněž stručné shrnutí zjištění určené vývojářům, kteří budou v práci na aplikaci pokračovat, aby se předešlo opětovnému řešení již identifikovaných problémů.

8.1.1 Návrhy k implementaci a dalšímu testování

Jedním z prvních kroků, které je třeba v budoucím vývoji dokončit, je kompletní implementace nápověd (*tooltips*) a to zejména při chybném zapojení prvků a u vybraných typů datových propojení. Během testování rovněž vzešel návrh na zavedení vyhledávání v rámci menu, které by umožnilo uživatelům snadno nalézt jednotlivé položky dle názvu. Podobné řešení využívají další programy pracující s uzly, jako Blender a Maya. Tato funkce by významně usnadnila práci zejména při vyhledávání mezi velkým množstvím operátorů. K přehlednosti operátorů by také mohlo přispět sjednocení uzlů operací nad jednotlivými datovými typy do jednoho univerzálního. V rámci uzlů by bylo rovněž vhodné lépe upozornit na možnost zobrazit při kliknutí na hodnotu nabídku často využívaných hodnot a nabídku zpřehlednit. Tím by se usnadnila orientace uživatelů a zrychlila práce s běžně používanými hodnotami. Další navrhovanou funkcí je automatické zarovnávání uzlů do mřížky (*snap to grid*), které by vedlo k lepší organizaci pracovního prostoru a zjednodušilo manipulaci s uzly. V rámci uživatelského rozhraní je připravena implementace ikonového menu, jehož dokončení závisí na výběru výstižných ikon pro jednotlivé položky a ověření jejich přehlednosti a použitelnosti skrze další testování. Dále by bylo vhodné umožnit vkládání objektů přímo do scény a zároveň prozkoumat možnost přidávání transformací přímo do uzlu sekvence nebo kamery, což by mohlo zefektivnit pracovní postupy uživatelů.

8.1.2 Oblasti ke zlepšení

V průběhu práce byly rovněž identifikovány oblasti aplikace, kterým by bylo vhodné se v budoucnu detailněji věnovat, například v rámci další bakalářské či diplomové práce.

Prvním z těchto témat je systém práce s úhly v prostředí I3T. V současné verzi jsou úhly reprezentovány jako desetinná čísla v radiánech, což se v rámci testování ukázalo jako uživatelsky nepřívětivé. Bylo by proto vhodné navrhnout srozumitelnější způsob jejich reprezentace, zadávání i používání, a to jak v rámci rotačních transformací, tak i v dalších částech programu. Tento výzkum by zároveň mohl být propojen s návrhem přehlednější vizualizace kvaternionů, která by studentům počítačové grafiky mohla pomoci lépe porozumět tomuto matematickému konceptu.

Další oblastí vhodnou k dalšímu rozpracování je validace uživatelského vstupu, zejména při zadávání hodnot. Současná implementace je nedostatečná – chybové

hlášení zobrazené v horní části uzlu je málo výrazné a uživatelé jej často přehlížejí. Bylo by tedy vhodné navrhnout a implementovat systém, který by uživatele srozumitelně informoval o výskytu chyby, včetně přesného označení místa, kde k ní došlo, a vysvětlení její příčiny. Tento návrh by pravděpodobně vyžadoval hlubší zásah do architektury aplikace, ale mohl by výrazně přispět ke zvýšení její použitelnosti. S validací vstupů úzce souvisí i druhý návrh, a to možnost nastavení povoleného rozsahu hodnot. Z tohoto důvodu je důležité navrhnout takový způsob zadávání a vizualizace těchto mezí, který bude pro uživatele srozumitelný a intuitivní.

Zásadní otázkou, která během práce vyvstala, je celkový koncept operátorů v aplikaci, který se pro uživatele ukázal jako matoucí. Jedním z hlavních problémů je skutečnost, že operátory nelze přímo připojovat k modelům ani je jednoduše vkládat do sekvence, což odporuje běžným uživatelským očekáváním. Nabízí se proto úvaha, zda by nebylo vhodné systém operátorů upravit. Jednou z možností je přiblížit je spíše podobě kalkulačky, tedy vytvořit z nich jednoduché uzly, které by se umísťovaly přímo na spojovací dráty a prováděly konverze a operace mezi daty. Alternativním řešením je zachovat současný koncept, ale výrazně zlepšit informovanost o jejich funkci a zjednodušit manipulaci s nimi, například intuitivnějším napojováním na modely.

Poslední problematikou, které je třeba se v budoucnu dále věnovat, je rozhraní cyklu, jenž v aktuální podobě působí na uživatele nepřehledně a mohutně, což je může odrážet od jeho použití. Bylo by vhodné navrhnout pro cyklus přehlednější a srozumitelnější způsob zobrazení, který by usnadnil jeho pochopení i práci s ním. S tím úzce souvisí i potřeba celkově promyslet systém animace v rámci aplikace, aby byl pro uživatele intuitivní a přirozeně zapadal do jejich pracovního postupu.

Kapitola 9

Závěr

Cílem této práce bylo zlepšit použitelnost výukové aplikace Interactive Tool for Teaching Transformations (I3T). V úvodní fázi byly zvoleny vhodné techniky pro analýzu a návrh vylepšení použitelnosti, konkrétně uživatelské testování a principy iterativního a paralelního návrhu. Uskutečněné testování s uživateli poskytlo cenné poznatky o aktuálním stavu aplikace, na jejichž základě byl sestaven přehled hlavních problémových oblastí, které vyžadovaly zlepšení.

V průběhu práce vzniklo množství návrhů na zlepšení použitelnosti, z nichž značná část byla postupně implementována týmem pracujícím na I3T. Tyto úpravy byly opakovaně ověřovány v několika kolech uživatelského testování. Během tohoto procesu došlo k dalšímu rozvoji návrhů, výběru nejvhodnějších variant, formulaci nových podnětů a současně i k opuštění některých původních řešení, která se v praxi neosvědčila.

Na základě celkového pozorování a vyhodnocení výsledků testování pomocí jednoduché bodovací metriky lze konstatovat, že došlo k výraznému zlepšení celkové použitelnosti aplikace i uživatelského zážitku, a to o 30 až 60% ve čtyřech z pěti klíčových kategorií. Vylepšené uživatelské rozhraní by mělo přispět k efektivnějšímu využití nástroje ve výuce a zároveň podpořit motivaci studentů k jeho používání.

Práce rovněž přinesla seznam doporučení pro budoucí vývoj nástroje. Tyto návrhy bude vhodné v dalším vývoji prozkoumat, implementovat a dále testovat s cílem zajistit co nejlepší uživatelský zážitek.

Literatura

- [1] Michal Folta. *Systém na výuku transformací. Diplomová práce. FEL, ČVUT v Praze.* 2016.
- [2] Jiří Boček. *Testování užitečnosti nástroje pro výuku transformací. Bakalářská práce. FEL, ČVUT v Praze.* 2024.
- [3] Jakob Nielsen. *Usability 101: Introduction to usability.* 2024.
<https://www.nngroup.com/articles/usability-101-introduction-to-usability/>. Datum poslední návštěvy: 23.4.2025.
- [4] Jakob Nielsen. *Iterative design of User Interfaces.* 2020.
<https://www.nngroup.com/articles/iterative-design/>. Datum poslední návštěvy: 24.4.2025.
- [5] Therese Fessenden. *Parallel & Iterative Design + competitive testing = high usability.* 2024.
<https://www.nngroup.com/articles/parallel-and-iterative-design/>. Datum poslední návštěvy: 24.4.2025.
- [6] Steve Krug. *Web design : nenutě uživatele přemýšlet!* Computer Press, 2006.
- [7] Jakob Nielsen. *Why you only need to test with 5 users.* 2024.
<https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>. Datum poslední návštěvy: 28.4.2025.
- [8] Elizabeth Goodman, Mike Kuniavsky a Andrea Moed. *Observing the User Experience, Second Edition: A Practitioner's Guide to User Research.* 2nd vyd. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2012. ISBN 0123848695.
- [9] Vít Zadina. *Testování užitečnosti nástroje pro výuku transformací. Bakalářská práce. FEL, ČVUT v Praze.* 2019.
- [10] Dan Rakušan. *Vylepšení uživatelského rozhraní aplikace I3T. Diplomová práce. FEL, ČVUT v Praze.* 2025.
- [11] *Connect nodes by creating connection lines.*
<https://help.autodesk.com/view/MAYAUL/2025/ENU/?guid=GUID-C6E5588F-40A4-4B1E-9C02-A29DA9A4650F>. Datum poslední návštěvy: 18.5.2025.
- [12] *Material Editor Guide.*
<https://dev.epicgames.com/documentation/en-us/unreal-engine/connecting-nodes-in-unreal-engine>. Datum poslední návštěvy: 19.5.2025.
- [13] *Connecting Nodes.*
<https://dev.epicgames.com/documentation/en-us/unreal-engine/unreal-engine-material-editor-user-guide>. Datum poslední návštěvy: 19.5.2025.
- [14] Katie Sherwin a Samhita Tankala. *Card sorting: Uncover users' mental models.* 2024.
<https://www.nngroup.com/articles/card-sorting-definition/>. Datum poslední návštěvy: 28.4.2025.

- [15] Matvei Korikov. *Dokovatelná okna, historie, logování, jazykové mutace a optimalizace nástroje I3T*. Bakalářská práce. FEL, ČVUT v Praze. 2025.

Příloha A

Scénáře testování

Příloha obsahuje testovací scénáře, které měli účastníci k dispozici během testování. Nezahrnuje scénář z testování s papírovými prototypy dne 18.3., jelikož v tomto případě probíhalo zadávání úkolů výhradně ústní formou z důvodu složitějšího způsobu vysvětlování práce s prototypem.

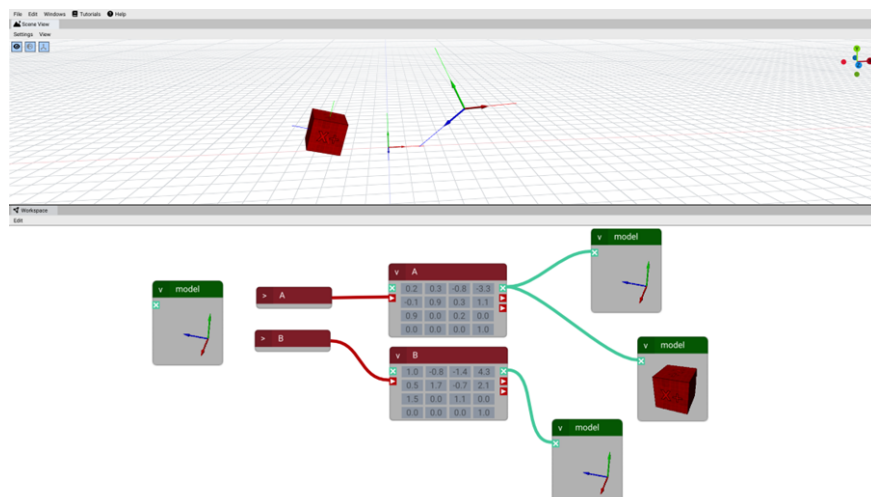
A.1 Úvodní testování 18.–22.11.

Vítej na testování použitelnosti aplikace I3T. Tento dokument tě provede celým testováním. Nejprve tě čeká krátký dotazník ohledně tvých zkušeností s aplikací. Na druhé straně najdeš 3 úkoly, které je třeba splnit v aplikaci. Během testování mluv nahlas, popisuj svoje myšlenkové pochody a komentuj, co děláš. Pro účely tohoto testování nepoužívej tutoriály aplikace. Nejedná se o zkoušku, netestuje tvoje znalosti látky, takže pokud nerozumíš úloze z pohledu počítačové grafiky, můžeš se ptát na jakékoliv otázky.

Otázky před začátkem:

- Už jsi viděl aplikaci I3T?
- Používal jsi již aplikaci I3T?
- Pokud jsi na některou z předchozích otázek odpověděl ANO, jaký byl tvůj první dojem z aplikace?

1. Úkol: Představ si, že zítra píšeš test z maticových transformací a právě jsi si přečetl, že na pořadí matic aplikovaných na objekt záleží. Nejsi si však jistý, jak to funguje v praxi. Naštěstí máš stažený program I3T a rozhodneš se tuto skutečnost otestovat přímo tam. *V programu vytvoř dva stejné modely (k jejich rozlišení můžeš použít barvy) a na oba aplikuj stejné transformace, ale v jiném pořadí a porovnej výsledky.*
2. Úkol: Zkouška dopadla skvěle, díky tomu, že jsi si tak dobře procvičil transformace. Nyní však čelíš jinému problému. Uzávěrka tvého semestrálního projektu v OpenGL se blíží a ty jsi do své scény stále neimplementoval kameru, protože si přesně nejsi jistý, jak to funguje a jak vypadají pohledové a projekční matice. Otevřeš I3T v naději, že lépe porozumíš tématu. *Do pracovní plochy přidej kameru, vlož transformace pohledu a projekce (view a projection), připoj k ní obrazovku a nastav hodnoty tak, aby byl alespoň jeden z dříve vytvořených objektů vidět na připojené obrazovce (screen).*
3. Úkol: Dostal jsi se až k závěrečné zkoušce. A snažíš se na ni připravit co nejlépe. Narazíš na úlohu, která obsahuje objekt v souřadnicovém systému definovaném jednou maticí a máš za úkol jej umístit do souřadnicového systému definovaného jinou maticí. Otevřeš program I3T, abys otestoval úlohu v praxi. *Otevři soubor task3.scene, tam najdeš dvě matice A a B a model transformovaný pomocí A. Tvým úkolem je zobrazit tento model v souřadnicovém systému, který je definován maticí B.*



Obrázek A.1. Scéna k úkolu 3.

A.2 Testování 15.-16.4.

Vítej na testování použitelnosti aplikace I3T. Tento dokument tě provede celým testováním. Čekají tě 3 úkoly, které je třeba splnit v aplikaci. Během testování mluv nahlas, popisuj svoje myšlenkové pochody a komentuj, co děláš. Pro účely tohoto testování nepoužívej tutoriály aplikace. Nejedná se o zkoušku, netestuje tvoje znalosti látky, takže pokud nerozumíš úloze z pohledu počítačové grafiky, můžeš se ptát na jakékoliv otázky. Na závěr vyplň krátký dotazník.

Představ si, že jsi dostal domácí práci z PGR, kterou musíš zítra odevzdat. Máš za úkol vytvořit scénu s kamerou a alespoň dvěma objekty, na které budou aplikovány transformace.

1. Úkol: *V programu vytvoř model a nejdříve ho otoč o 30° podél osy x a následně posuň o vektor $(1, 2, 3)$. Dále změň jeho barvu a počet desetinných míst jednotlivých matic na 2.*
2. Úkol: *Do workspace přidej kameru, přidej transformace pohledu a projekce (view a projection), připoj k ní obrazovku a pohraj si s hodnotami, abyste viděli objekt na připojené obrazovce.*
3. Úkol: *Vytvoř ještě jeden model a aplikuj na něj inverzi transformací aplikovaných na původní model.*

Dotazník na závěr:

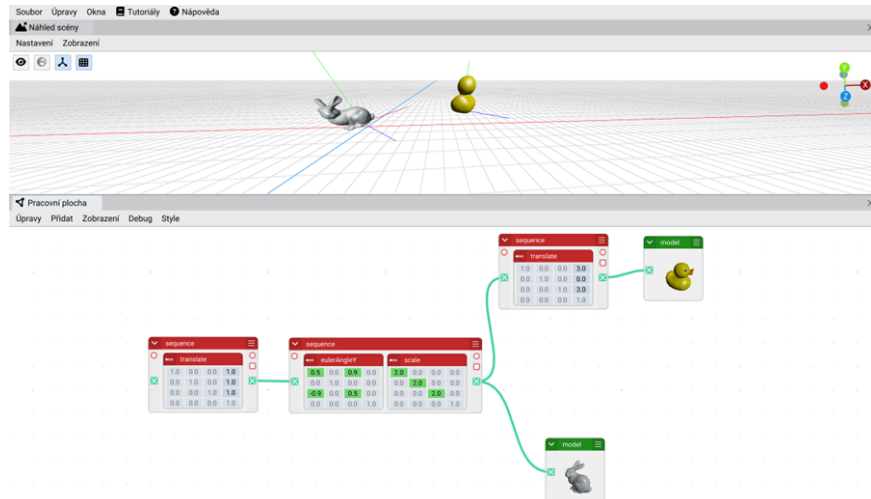
- Který typ pinů (zapojení) ti přijde nejlepší? Popiš.
- Který typ nastavení režimu zobrazení uzlu ti přijde nejlepší? Popiš.
- Prostor pro další připomínky, komentáře.

A.3 Testování 7.5

Vítej na testování použitelnosti aplikace I3T. Tento dokument tě provede celým testováním. Čekají tě dva úkoly, které je třeba splnit v aplikaci. Během testování mluv nahlas, popisuj svoje myšlenkové pochody a komentuj, co děláš. Nejedná se o zkoušku, netestuje tvoje znalosti látky, takže pokud nerozumíš úloze z pohledu počítačové grafiky, tak se ptej na jakékoliv otázky.

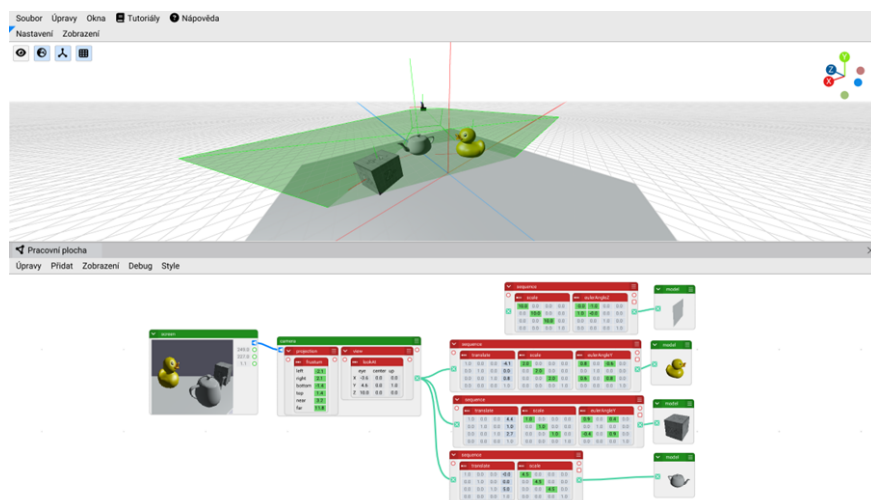
Představ si, že se připravuješ na zkoušku z PGR. Stále je několik principů, které ti dělají problém. Hlavně rozdíl v pořadí aplikace transformací na model a řetězec transformací aplikovaných na model při jeho zobrazování na obrazovku.

1. Úkol: Nejprve si chceš procvičit rozdílné pohledy na aplikování transformací na model. Z pohledu transformace modelu a z pohledu transformace soustavy souřadnic. *V otevřené scéně programu vytvoř model, aplikuj na něj dvě libovolné transformace a zapoj ho do připraveného řetězce transformací. Spust krokování zleva a potom zprava a pozoruj rozdíl.*



Obrázek A.2. Scéna k úkolu 1.

2. Úkol: Aplikování transformací ti už dává smysl. Tak je čas posunout se s učením dál. Chtěl bys pochopit jak funguje celý řetězec transformací aplikovaných na model při jeho zobrazování na obrazovku. *V otevřené scéně spust krokování. A projdi celý řetěz transformací. V scene view se pokus replikovat to co se zobrazuje v uzlu obrazovky (screen).*



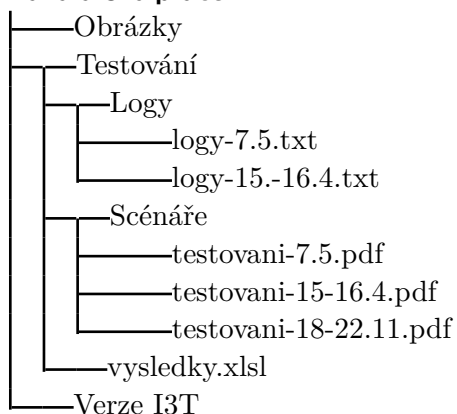
Obrázek A.3. Scéna k úkolu 2.

Příloha B

Obsah elektronické přílohy

Celý souborový systém elektronické přílohy je vyobrazen na diagramu níže.

Bakalářská práce



Složka *Obrázky* obsahuje veškeré obrázky použité v bakalářské práci, adresář *Testování* zahrnuje složku se záznamy uživatelské aktivity, které byly vygenerovány přímo programem I3T, složku se scénáři testování, a tabulku obsahující data a vyhodnocení výsledků z úvodního testování a druhé etapy testování a složka *Verze I3T* obsahuje dvě verze I3T: v2.0.0-rc3, která zachycuje stav programu před začátkem práce a verzi release-new, která není sama o sobě release verzí, ale reprezentuje stav programu na konci práce. Ke spuštění programu stačí vyextrahovat .zip soubor a z rozbaleného adresáře spustit program I3T.exe.