

Samostatný projekt - Návrh scénářů pro výuku
transformací s podporou I3T

Adam Loucký

Leden 2023

1 Úvod

1.1 Abstrakt

Tato práce se zabývá procesem vývoje tutoriálů pro výukový program I3T. K jejich tvorbě byla využita znalost metod výuky neznámých konceptů pomocí tutoriálů a úkolů. Efektivita tutoriálů byla později ověřena testováním.

1.2 Program I3T

Program I3T slouží k výuce transformací ve 3D prostoru. K interakci s 3D scénou využívá tzv. bloků, které slouží jako základní jednotka programu. Bloky se mezi sebou propojují, tím se tvoří 3D scéna.

Starší verze programu byla využívána v předmětu Počítačová Grafika k vizuální demonstraci základních konceptů předmětu. Nová verze I3T vznikla za účelem zpříjemnění uživatelského rozhraní a zpřístupnění programu dalším uživatelům.

V nové verzi programu však nebyla aktualizována dokumentace ani návody pro uživatele. Vznikla tedy potřeba vytvořit tutoriály nové, tak, aby přehledně a srozumitelně popsaly, jak s programem pracovat.

Nejprve bylo nutné se s programem seznámit. Následně byly navrženy nové tutoriály, jejichž cílem je zejména umožnit pracovat s programem studentům, ale také lidem z řad laické veřejnosti, kteří se s podobným programem nesetkali.

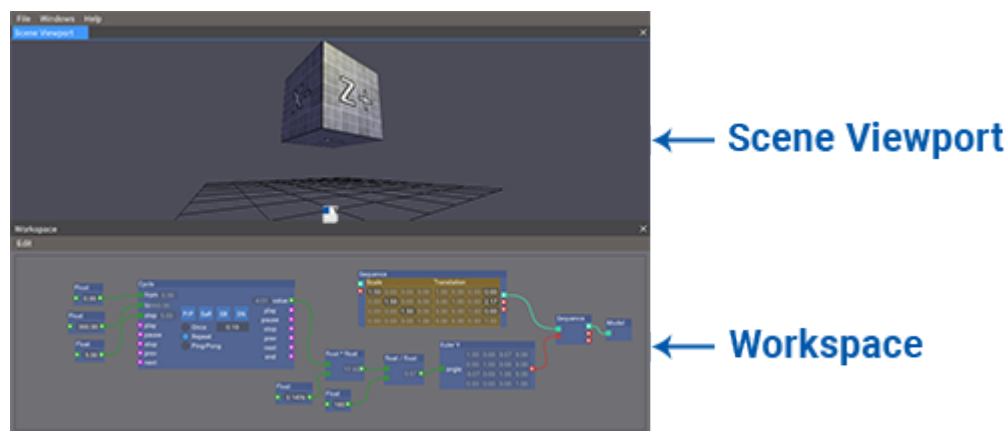


Logo programu I3T

2 Seznámení s výukovým programem I3T

2.1 Seznámení s starými tutoriály

Po vstupu autora do vývojářského týmu bylo nutné seznámit se se současným stavem programu a tutoriálů. Tato část projektu trvala přibližně tři týdny. Program obsahoval dva nedokončené tutoriály, které vypracoval student, který dříve zpracovával grafický vzhled programu. Tyto tutoriály však sloužily pouze pro rychlé seznámení studenta s programem a v době zahájení tohoto samostatného projektu již nebyly aktuální vzhledem k současnému stavu vývoje programu.



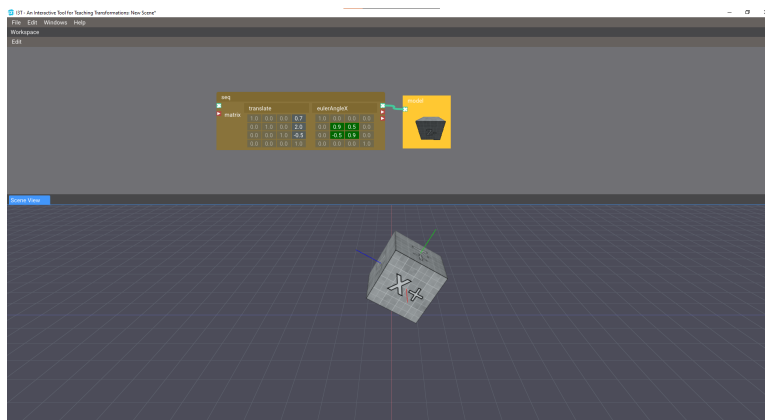
Obrázek zastaralé scény

Instrukce obsažené v tutoriálech byly nejasné a nezodpovídaly dotazy, které by uživatel mohl mít. Také nebyly přizpůsobené aktuálním verzím nástrojů používaných při tvorbě scény.

Tutoriály byly psané v upraveném jazyce Markdown, který program načítá do odděleného okna a formátuje ho. Bylo tedy nutné se seznámit se i s jazykem Markdown a konvencemi používanými ve starších tutoriálech.

2.2 Seznámení s programem

V další fázi projektu bylo nutné seznámit se se samotným programem. Rozhraní se skládá ze dvou oken, **Workspace** a **Scene View**.



Workspace na vrchní polovině, Scene View na spodní polovině

Workspace slouží k interakci se scénou pomocí stavebních bloků, které mezi sebou propojujeme. Bloky můžeme ve workspace přidávat i odebrat pomocí pravého tlačítka myši.

Scene View ukazuje scénu z nezávislé kamery. Kamerou lze otáčet, posouvat a měnit její vzdálenost od středu scény pomocí tažení myši.

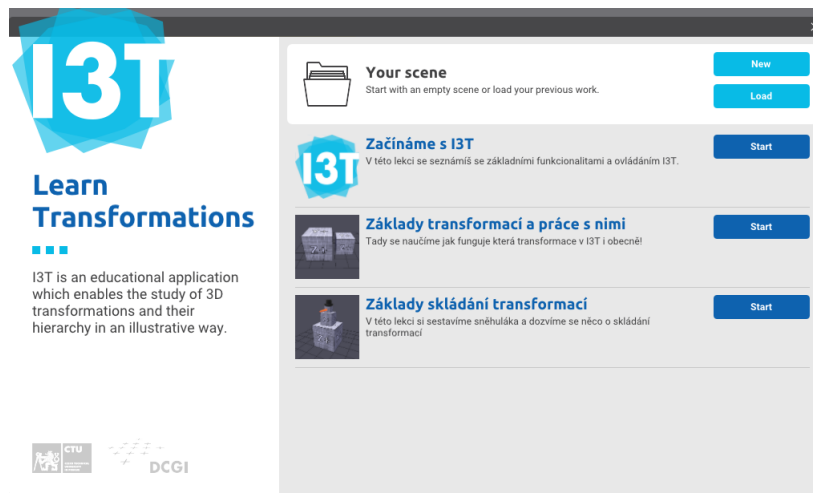
Dále má program vrchní lištu se záložkami **File**, **Edit**, **Windows** a **Help**.



Horní lišta programu

- **File** slouží k ukládání a načítání scén
- **Edit** k vracení změn
- **Windows** ke správě oken programu
- **Help** k dodatečné dokumentaci a přístupu do demo verze programu

K tutorialům samotným má uživatel přístup ze startovního okna, kde je výčet všech vytvořených tutorialů. Po spuštění jednoho z nich se otevře okno, přes které může student tutorial procházet.



Úvodní menu programu

2.3 Rešerše: obsah tutoriálů

Na tvorbu obsahu tutoriálů měly vliv zdroje, které jsou dostupné na internetu. Mezi nejzajímavější zdroje patří "3D Transformations" od Østfold University College nebo prezentace předmětu Programování grafiky (*Transformations (in OpenGL)* od Petra Felkela a Jaroslava Sloupa 2021)

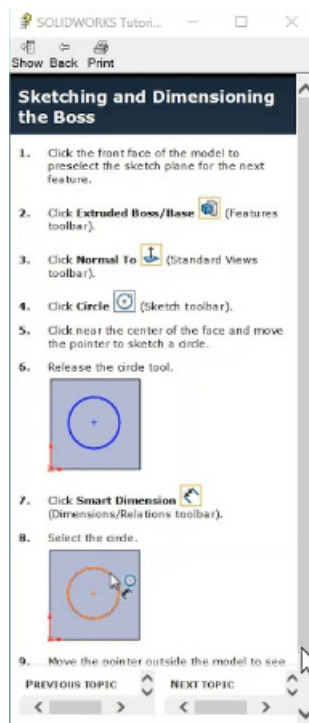
"3D Transformations" je více stručný a přímý zdroj. Jednotlivé transformace jsou vysvětleny pouze velmi krátce, více pozornosti ale stránka věnuje využití transformací. Vysvětluje skládání transformací a také ukazuje konkrétní využití transformací v OpenGL. Také se zde objevují často netradiční transformace, jako například zrcadlení nebo shear.

Transformations (in OpenGL) slouží spíše k teoretickému pochopení transformací, a věnují se tématu více do hloubky. Důležité je, že přímo ukazují spojení mezi látkou Lineární algebry a Počítačové grafiky.

Tyto zdroje sloužily jako inspirace celkového obsahu tutoriálů. Jasně definují termíny týkající se 3D grafiky a ukazují jak tyto termíny srozumitelně vysvětlit.

2.4 Rešerše: grafické rozhraní

Grafické rozhraní tutoriálového okna bylo vytvořeno v rámci bakalářské práce "*Testování užitečnosti nástroje pro výuku transformací*" (Vít Zadina 2020). Obecně se forma tutoriálového okna podobá tutoriálu modelovacího programu Solidworks. Solidworks se podobá programu I3T i náhledem na scénu a zaměřením na 3D prostor.



Rozhraní tutoriálu v Solidworks

Tématem tutoriálů Solidworks je tvorba modelů. Témata, kterými se tutoriály zabývají jsou vždy rozdělené na jednotlivé stránky. Stránky s zároveň formátují dle velikosti okna tak, aby text zůstal čitelný.

Uživatel tak vidí vše, co potřebuje a nemusí se vracet na předchozí stránky. Tutoriálové okno solidworks má nad nadpisy horní lištu, která obsahuje několik tlačítek s užitečnými funkcemi při procházení tutoriály.

Dalším programem, který se I3T podobá náhledem na 3D scénu je program Blender. Blender žádný programový tutoriál nemá. Namísto toho využívá tutoriálových videí, pomocí kterých učí základní koncepty programu. Videá mají oproti textovým tutoriálům významnou výhodu, mohou totiž ukázat postupy vizuálně.

Tyto poznatky pak aplikujeme na I3T:

- dělení obsahu na stránky podle témat
- formátování textu podle velikosti okna
- přidání tlačítek pro užitečné funkce
- přidání .gif animací do tutoriálů

2.5 Ostatní práce na programu

Seznámení s programem také probíhalo řešením dílčích problémů a plněním úkolů, jejichž vyřešení později usnadnilo práci na uživatelských návodech. Úkoly, které bylo nutné vypracovat, se týkaly zejména:

- načítání tutoriálových souborů
- částečné úpravy barevných konstant za účelem lepšího stylování programu
- implementace funkce Copy+Paste

Také bylo nutné věnovat se stavu grafického rozhraní a jeho vlivu na interakci s tutoriály. V programu totiž nebyl žádný ověřovací systém, podle kterého by bylo možné kontrolovat stav úkolu, zadaného uživateli. Zajištění korektnosti uživatelského řešení bylo tedy nutné koncipovat jinak.

Pro účely kontroly bylo do tutoriálů přidáno velké množství obrázků zachycujících stav programu, který odpovídá správnému řešení. Uživatel si tak může ověřit, zda úkol splnil nebo ne.

Toto triviální řešení by později mohlo být nahrazeno interaktivnější variantou, která by kontrolovala stav modelů a bloků ve scéně nebo uživateli napovídala přímo ve Scene View.

3 Tvorba tutoriálů

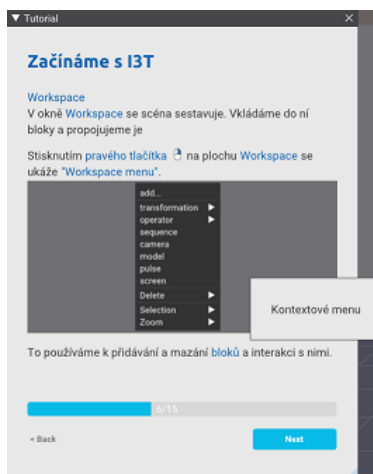
3.1 Jazyk tutoriálů

Cílovou uživatelskou skupinou tohoto projektu jsou studenti předmětu "Programování grafiky", což je předmět vyučovaný v českém jazyce. Tutoriály jsou proto zpracovány také v češtině. Čeština zároveň umožňuje srozumitelněji vysvětlit složitější koncepty.

3.2 Nástroje tvorby tutoriálů

Tutoriály využívají těchto nástrojů:

- **Task** - úkol, který má uživatel splnit
- **Tip** - prokliknutelné tlačítko, které zobrazí nápovědu
- **Tooltip** - popisek, který se zobrazí při výběru obrázku v tutoriálu



Tooltip - vyskakující nápověda

Tyto nástroje slouží ke zlepšení interaktivity tutoriálů, poskytují uživateli dodatečné informace, radí mu a říkájí mu co má dělat. Nástroje však bylo nutné aktualizovat, nefungovaly totiž tak, jak by měly.

Tooltips zobrazovaly pouze jméno obrázku a text v tooltipu byl nečitelný. Tipy byly také špatně čitelné. Samotný text tutoriálu se při zvýraznění stal světle šedivým. To paradoxně snižovalo výraznost textu.

Všechny tyto problémy byly vyřešeny v rámci úvodní práce na programu - Tooltips nyní zobrazovaly krátký popis obrázku a Tipy a zvýrazněný text byly čitelnější.

3.3 Princip tvorby tutoriálů

V tutoriálech se objevují tři hlavní metody.

První metoda je připodobnění.

"When students study new concepts, meaningful learning proceeds when they find and visualise connections between a newly taught context and what they already know." (Harrison, A.G., Treagust, D.F. 2006).

Příkladem může být **Scene View** v I3T. **Scene View** slouží jako náhled do 3D scény kterou uživatel vytváří. Připodobněním pak může být **Object mode** v programu Blender, což je program, který studenti oboru Počítačová grafika používají, a znají. Interakce s oběma náhledy je velmi podobné, student se tedy zároveň naučí ovládání i pochopí funkci našeho **Scene View**.

Druhou metodu je výuka pomocí úkolů, neboli Task Based Learning (*viz Büyükkarçı, Kağan. "A critical analysis of task-based learning." (2009)*).

V našem případě to znamená zadávat uživateli úkoly během lekce. Interakce s programem změní tempo tutoriálu z pasivního čtení na aktivní řešení problému. Tím se vrátí pozornost uživatele zpět k programu a sám si může vyzkoušet nově nabyté vědomosti na úkolu.

Třetí metodou je separace jedné komplexní úlohy na co nejjednodušší podúlohy. Pokud uživateli zadáme jednu složitou úlohu můžeme narazit na problémy. Komplexní úkol znemožňuje jednoduše a efektivně uživateli poradit, protože nevíme, na které části úkolu právě pracuje. Také musíme před komplexní úlohou vysvětlit větší množství konceptů, které musí studenti všechny udržet až do zadání úkolu a hrozí že studenty zahltime informacemi. Úlohu tedy rozdělíme na menší podúlohy, které jsou stravitelnější a rozprostřeme je mezi pasivní části tutoriálu.

Tato metoda je nejčastěji používána v počítačových hrách, kde hráče seznamujeme s koncepty postupně v průběhu hry. Tuto metodu zmiňuje přednáška Asher Vollmera "How to Make Great Game Tutorials". V přednášce se pojednává o důležitosti zábavnosti tutoriálů a významu dělení informačních částí na menší, stravitelnější části.

Kombinací těchto metod vznikla šablona, podle které byly následně jednotlivé tutoriály vytvářeny:

1. **Vysvětlíme, čeho se lekce týká**
2. **Vysvětlíme teorii potřebnou splnění dalšího okamžitého podúkolů**
3. **Zadáme podúkol, který využívá nově získané informace**
4. **Opakujeme od bodu 2. dokud nevysvětlíme vše potřebné**
5. **Shrnutí lekce**

3.4 První tutoriál: Začínáme s I3T



Náhled prvního tutoriálu

Starý úvodní tutoriál se věnoval rozhraní a ovládání programu pouze stručně. Úvodní instrukce byly tedy doplněny a aktualizovány tak, aby se o programu uživatel dozvěděl co nejvíce, ale zároveň nebyl zahlcený informacemi. Tutoriál je úvodem do programu, je tedy nutné použít odlišnou strukturu:

- vysvětlení základních termínů
- ukázka rozhraní programu
- jednoduchý úkol

Nejprve je třeba seznámit se s tutoriálovou terminologií, tedy s tím co znamená **Tip**, **Task** a **Tooltip**.

Dále se tutoriál věnuje popisu rozhraní programu, které je uživateli vysvětleno spíše stručněji. Zachází s rozhraním si studenti osvojí v průběhu tutoriálu prostřednictvím samotné interakce s rozhraním.

Uživatel má na konec za úkol přidat tři bloky a korektně je propojit podle instrukcí. Uživatel se tak blíže naučí pracovat s bloky, ale také si osvojí základy transformací 3D objektů. Zároveň interaguje s programem, čímž se učí pracovat i s jeho rozhraním.

Úkol zadaný uživateli byl v originální verzi mnohem komplexnější a vyžadoval přidání velkého množství bloků. Složitost i rozsah úkolu se výrazně snížily, aby uživatele tutoriál nefrustroval komplikovanými úlohami už v první lekci.

Zároveň se v tutoriálu využívaly operátory. Operátor je blok, který se funkcí podobá sekvenci, je však méně názorný a neumožňuje vkládání několika transformací, což je potřeba v pozdějších tutoriálech. Část tutoriálu s operátory byla tedy vynechána.

3.5 Druhý tutoriál: Základy transformací



Náhled druhého tutoriálu

Tématem druhého tutoriálu jsou transformace v 3D prostoru. Obdoba tohoto tutoriálu v původní verzi nebyla, bylo tedy nutné vytvořit kompletně novou lekci.

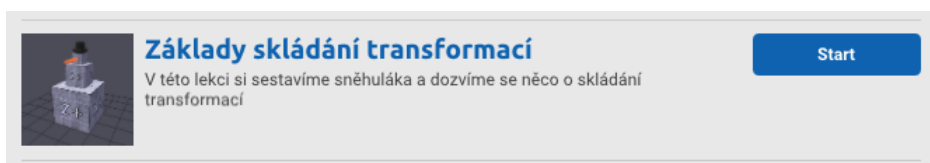
Velkou překážkou během tvorby tutoriálu bylo množství teoretických informací potřebných k pochopení konceptu transformací. Vysvětlování jakéhokoliv problému je náročné koncipovat tak, abychom udrželi pozornost uživatele. Bylo tedy nutné množství teorie zredukovat na co nejmenší, zároveň však zachovat všechny důležité informace a neobcházet přesně dané definice.

Východiskem tedy bylo formulovat tutoriál jako postup řešení problému. Problémem bylo v našem případě "Nevím jak aplikovat translaci na objekt." Uživateli je vysvětleno co to je transformace, homogenní souřadnice a jak s homogenními souřadnicemi pracovat. Tím se dozví, jakým způsobem probíhá transformace z teoretické stránky. Zároveň vidí i využití nově naučené látky.

Následuje praktická část lekce, kdy uživatel využívá nabitých znalostí k transformaci modelu rychle různými druhy transformací. Tato část slouží jako opakování základů ovládnutí programu a zároveň vizualizuje, jak fungují jednotlivé transformace.

3.6 Třetí tutoriál: Základy skládání transformací

Třetí tutoriál se soustředí primárně na využití znalostí nabytých v předchozích lekcích, sekundárně pak na teorii skládání transformací.

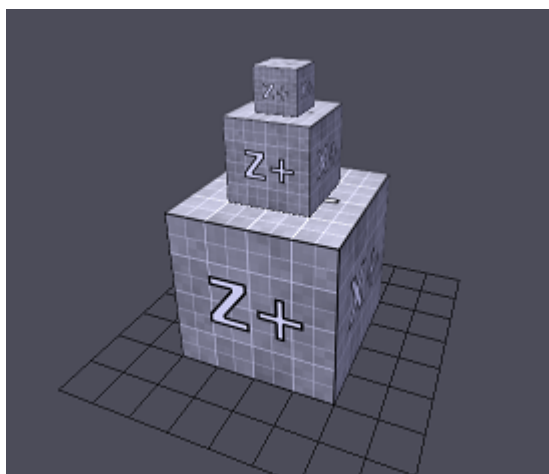


Náhled třetího tutoriálu

Dodatečně také prohlubuje schopnost pracovat s I3T jako takovým (učí studenta jak skládat transformace v programu I3T). Lekce samotná obsahuje méně teorie, je tedy tvořená hlavně úkoly.

Celkovým motivem třetího tutoriálu je stavba sněhuláka. Inspirací pro tento tutoriál byla předchozí verze návodu, která učila skládání transformací podobným způsobem. Rozhraní programu se však ve starších verzích I3T lišilo, bylo tedy nutné starý tutoriál aktualizovat. Nový tutoriál přebírá motiv stavby sněhuláka, postup je však jiný.

Uživatel si nejdříve přidá nohy sněhuláka do scény, na nich si názorně vyzkouší důsledek prohození pořadí transformací, který je popsán i teoretickou vsuvkou. Později do scény přidá i tělo a hlavu, jejichž velikost upraví podle instrukcí a přesune je tak, aby připomínaly sněhuláka.



Výsledný "sněhulák"

4 Testování a ohlas

V rámci testování v průběhu tvorby lekcí byli osloveni tři dobrovolníci z řad studentů Českého vysokého učení technického v Praze k otestování programu a tutoriálů. Testování se účastnili i dva laikové. Testujícím byl autor této práce.

4.1 Postup testování

Každému uživateli byla zaslána spustitelná verze programu. Jejich úkolem bylo program spustit, otevřít si první tutoriál a dokončit ho.

V průběhu testování uživatel sdílel svou obrazovku. Testovaným bylo dále doporučeno přemýšlet nahlas, tedy říkat nahlas vše, co je napadne. Testovaný měl za úkol uživatele sledovat a zapisovat si jeho chování.

Po absolvování prvního tutoriálu obdrželi testovaní úkoly, který ověřovaly, zda si pamatují koncepty z tutoriálu. Úkoly zněly následovně:

- "Pootoč kamerou v Scene View."
- "Jak bys přidal další matici transformace?"
- "Dokázal bys odpojit sekvenci od modelu?"

Po splnění dodatečných úkolů test skončil. Testovaní byli dále dotázáni na hodnocení tutoriálů a programu jako takového.

4.2 Zaujatost ohlasu

S ohlasem a testováním přichází současně i problém se zaujatostí a validitou výsledků testů. Nutné je také zhodnotit, kdo je cílovou uživatelskou skupinou, od těchto uživatelů je pak ohlas a hodnocení nejdůležitější.

V našem případě to jsou hlavně studenti technických oborů, kteří mají nějaké předchozí znalosti matematiky a informatiky.

V případě tohoto testování však byli skoro všichni testovaní studenti ČVUT. Zbylí testovaní, kteří nejsou studenty technických oborů, měli velmi podobná hodnocení, jako ostatní respondenti, zaujatost výsledků tedy nehrozí.

4.3 Individuální testování

Feedback, který se týkal tohoto projektu je dělen na následující kategorie:

- Grafické změny
- Tutoriál chyby - text
- Tutoriál chyby - UI

Grafickou stránku programu hodnotili testovaní kladně. Mezi výhrady k rozhraní patřily hlavně detaily a menší změny, například "zvýraznění barvy zvýrazněného textu", "méně výrazný rámeček vybraného bloku" nebo "světlejší text v záhlaví tutoriálového okna". Tyto změny byly jednoduše napravitelné. Vnímání samotné grafické stránky programu je ale často subjektivní, proto byly více radikální vizuální změny z individuálního feedbacku upožděny dokud nebylo množství odpovědí větší.

Tutoriálové textové chyby byly lehce napravitelné, např jednoduché hrubky, pravopisné chyby, nebo nejasné věty. Nejčastějším problémem bylo nekonzistentní pojmenování oken, během vývoje tutoriálů se totiž název hlavního náhledového okna měnil ze "Scene Viewport" na "Scene View" . Všechny chyby byly opraveny okamžitě po odhalení.

Tutoriálové chyby v rozhraní byly nejzávažnější chyby, které se vztahovaly k funkcionalitě a rozhraní tutoriálů. Tyto chyby byly tři:

- chybějící tlačítko pro návrat do tutoriálového menu
- při spuštění nového tutoriálu zůstávají ve Workspace bloky
- chybějící blokace rozhraní programu dokud je zobrazené startovní okno

Upozornění na tyto nedostatky bylo nejdůležitější a nejvíce přínosnou informací z celého testování. Podobné chyby totiž matou uživatele a znemožňují mu aplikaci komfortně používat. Feedback velmi významně přispěl k vývoji a tvorbě tutoriálů a byl jednou z nejvýznamnějších částí celého projektu.

4.4 Dodatečné testování

Program byl dále testován v rámci předmětu "Uživatelský výzkum" studenty Vysoké školy ekonomické v Praze. Testovanými byli čtyři absolventi bakalářského studia na Českém vysokém učení technickém v Praze.

Feedback je dělen podobně jako v předchozí kapitole, na tři kategorie.

Uživatelé na grafické stránce programu hodnotili negativně hlavně barvu zvýrazněného textu, který připomíná hypertextový odkaz.

V tutoriálech se nacházelo několik zastaralých obrázků. Ty ukazovali jiné rozložení obrazovky, než je původní, což uživatele mátló. Také se maticové násobení v tutoriálech nesprávně označuje jako datový typ.

Jako velkou chybu rozhraní tutoriálů hodnotili uživatelé nevýrazné tlačítko Tip, které se často ztrácelo na stránkách s větším obsahem informací.

4.5 Individuální ohlas

Každý z testovaných byl konci testu dotázán na celkový dojem z programu a tutoriálů. Celkové hodnocení programu byly neutrální. Testování kritizovali hlavně blok sekvence, který byl podle nich neintuitivní. Testování naopak pozitivně hodnotili náhled na scénu, který je dostatečně názorný a poskytuje včasný feedback na akce uživatele.

Po seznámení s programem si však na nedostatky většinou zvykli a nijak jim nebránily v dalším zacházení s programem.

Recenze tutoriálů byly z větší části pozitivní. Testovaným vyhovoval neformální přístup a střídme výukové tempo. Také jim vyhovovaly obrázky v tutoriálech, které dodávaly potřebný kontext k instrukcím.

Kvalita tutoriálů je však značně omezená dostupnými prostředky pro jejich tvorbu.

Dalším výrazným zlepšením by tedy mohly být dodatečné funkcionality, které zlepší interakci s programem. Takovými funkcionality pak může být například "Tracking", který postupně aplikuje jednotlivé matice na model, přidání krátkých animací do tutoriálů místo obrázků nebo skriptování, které by umožnilo ovládat scénu přímo z Markdownových souborů.

5 Závěr

Tento samostatný projekt se zabýval procesem vývoje tutoriálů pro výukový program I3T. K tvorbě tutoriálů byla využita znalost metod výuky neznámých konceptů pomocí tutoriálů a úkolů.

Cílem bylo vytvořit tutoriály, které by pomohly studentům seznámit se s programem a transformacemi.

Efektivita tutoriálů byla na závěr ověřena testováním. Vzniklé tutoriály byly pozitivně hodnoceny jak ze strany studentů technických oborů, tak ze strany zástupců laické veřejnosti.

Cíl tohoto samostatného projektu byl splněn.

6 Seznam zdrojů

- BÜYÜKKARCI, Kağan. A critical analysis of task-based learning. 2009.
- Blender Fundamentals - https://www.youtube.com/watch?v=MF1qEhBSfq4list=PLa1F2ddGya_UvuAqHAksYnB0qL9yWDO6ab_channel=Blender
- HARRISON, Allan G. a David F. TREAGUST. Teaching and Learning with Analogies. In: AUBUSSON, Peter J., Allan G. HARRISON a Stephen M. RITCHIE, ed. Metaphor and Analogy in Science Education [online]. Berlin/Heidelberg: Springer-Verlag, 2006, s. 11-24 [cit. 2023-01-16]. Science Technology Education Library. ISBN 1-4020-3829-1. Dostupné z: doi:10.1007/1-4020-3830-5_2
- How to Make Great Game Tutorials - GDC https://www.youtube.com/watch?v=Uf7xLHUpKHEab_channel=GDC
- Solidworks built-in tutorials -<https://blogs.solidworks.com/solidworksblog/2013/05/get-more-with-solidworks-tutorials.html>
- Spatial Transformations by Dep. for Information Technology, Østfold University College <http://www.it.hiof.no/borres/j3d/math/threed/p-threed.html>
- TRANSFORMATIONS (IN OPENGL) Petr Felkel, Jaroslav Soukup - Department of Computer Graphics and Interaction, FEL Czech Technical University in Prague
https://cent.felk.cvut.cz/courses/PGR/lectures/04_Transform_1.pdf
https://cent.felk.cvut.cz/courses/PGR/lectures/05_Transform_2.pdf
- ZADINA, Vít. Testování užitečnosti nástroje pro výuku transformací. 2020.