



Zadání bakalářské práce

Název:	Prezentační systém v Blenderu
Student:	Matyáš Sojka
Vedoucí:	Ing. Lukáš Bařinka
Studijní program:	Informatika
Obor / specializace:	Webové a softwarové inženýrství, zaměření Počítačová grafika
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	do konce letního semestru 2022/2023

Pokyny pro vypracování

Analyzujte existující prezentační systémy z pohledu možností tvorby prezentací využívajících 3D a možností strukturování prezentací.

Analyzujte potřeby uživatelů pro tvorbu a ovládání prezentací, především s ohledem na multimediální a 3D obsah.

Navrhněte prezentační systém umožňující tvorbu multimediálních prezentací využívajících 3D založený na prostředí aplikace Blender.

Implementujte a otestujte navržený systém od vytváření prezentací, až po samotné prezentování obsahu.

Proveďte možnosti automatického generování takových prezentací.



**FAKULTA
INFORMAČNÍCH
TECHNOLÓGIÍ
ČVUT V PRAZE**

Bakalářská práce

Prezentační systém v Blenderu

Matyáš Sojka

Katedra softwarového inženýrství
Vedoucí práce: Ing. Lukáš Bařinka

13. května 2021

Poděkování

Rád bych poděkoval vedoucímu mé práce, Ing. Lukáši Bařinkovi, za jeho cenné a věcné rady a vedení v průběhu tvorby této práce. Poděkování patří i vybraným dobrovolníkům, kteří se zúčastnili uživatelského testování. V neposlední řadě bych chtěl poděkovat přátelům a rodině za ochotu a neustálou podporu.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 2373 odst. 2 zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů, tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu) licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 13. května 2021

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2021 Matyáš Sojka. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.

Odkaz na tuto práci

Sojka, Matyáš. *Prezentační systém v Blenderu*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2021.

Abstrakt

Prezentační systémy jsou v současné době velice rozšířenými nástroji. Umožňují jak vytváření prezentací, tak samotné prezentování. Téměř výhradně se současná řešení omezují na 2D prezentace.

Tato práce je určena uživatelům, kteří hledají alternativu k zevšedněným dvourozměrným prezentacím. Práce se zabývá návrhem a implementací prezentačního systému, který nabízí využití třetího rozměru. Dosahuje toho pomocí rozšíření (plug-in) aplikace Blender. Výsledné spojení plug-inu a prostředí Blenderu umožňuje vytváření dílčích 3D scén (komponent), jejich spojování do výsledné prezentace a následné odprezentování.

Klíčová slova plug-in, Blender, prezentační systém, 3D prostředí, Python

Abstract

Presentation software is nowadays a greatly widespread tool. It allows both the creation of presentations and the actual presentation. Almost exclusively, current solutions are limited to 2D presentations.

This thesis is intended for users who are looking for an alternative way to the commonly used two-dimensional presentations. The thesis includes the design and implementation of a presentation system that offers the use of the third dimension. It does that by using a Blender plug-in. Thanks to the plug-in and the Blender environment users can create 3D scenes (components), connect them into the final presentation and present them.

Keywords plug-in, Blender, presentation software, 3D environment, Python

Obsah

Úvod	1
Představení problému	1
Cíle a motivace práce	2
Struktura práce	2
1 Analýza	5
1.1 Analýza současných řešení	5
1.1.1 Existující prezentační systémy	6
1.1.1.1 Lineární prezentační systémy	6
1.1.1.2 Nelineární prezentační systémy	7
1.1.1.3 Video prezentační systémy	7
1.1.1.4 Trojrozměrné prezentační systémy	8
1.1.2 Potřeby prezentačního systému	8
1.1.2.1 Naplnění potřeb v rámci jednotlivých skupin	9
1.1.3 Možnosti aplikace Blender a jeho plug-inů	10
1.1.3.1 Naplnění minimálních potřeb prezentačního SW	10
1.1.3.2 Naplnění optimálních potřeb prezentačního SW	11
1.1.3.3 Existující plug-iny	12
1.2 Datová analýza	12
1.2.1 Struktura	13
1.2.1.1 Struktura prezentace	13
1.2.1.2 Struktura slidu	13
1.2.2 Obsah	14
1.2.2.1 Text	15
1.2.2.2 Multimédia	15
1.2.2.3 3D modely	15
1.2.3 Vizuální podoba	16
1.2.4 Rozdíly 2D a 3D prezentace	16
1.3 Analýza uživatelů	18

1.3.1	Uživatelské role a jejich potřeby	18
1.3.2	Případy užití	21
2	Návrh	29
2.1	Zavedení pojmů	31
2.2	Komponenty prezentace	31
2.3	Prezentace	33
2.4	Prezentování	36
2.5	Datový a stavový model	36
3	Implementace	41
3.1	Komponenty	42
3.1.1	Slidy	42
3.1.2	Generování	43
3.2	Prezentace	44
3.2.1	Vložení komponent	44
3.2.2	Změny v prezentaci	47
3.3	Prezentování	48
3.3.1	Ovládání prezentace	50
3.4	Problémy a omezení	50
3.5	Možnosti rozšíření	51
4	Testování	53
4.1	Pokrytí případů užití	53
4.2	Uživatelské testování	56
	Závěr	67
	Literatura	69
	A Slovník pojmů	75
	B Seznam použitých zkratk	77
	C Ukázky z výsledného prezentačního systému	79
	D Obsah příloženého CD	83

Seznam obrázků

1.1	Šablony rozložení prezentačního systému PowerPoint	14
1.2	Vizuální změna šablony v reakci na její strukturu – Google Slides .	17
1.3	Proces vytváření prezentace	19
1.4	Případy užití prezentačního systému v Blenderu	22
1.5	UC2 – Naplnění šablony daty	25
1.6	UC3 – Spojení slidů do prezentace	26
2.1	Návrh prezentačního systému v Blenderu	30
2.2	Zavedené pojmy pro prezentaci ve 3D	32
2.3	Návrh UI pro vytváření a generování komponent	33
2.4	Návrh layoutu pro vytváření a generování komponent	34
2.5	Návrh UI pro skládání a úpravy prezentace	35
2.6	Návrh layoutu pro skládání a úpravy prezentace	35
2.7	Návrh UI pro prezentování	37
2.8	Návrh layoutu pro posluchače	37
2.9	Návrh layoutu pro prezentujícího	38
2.10	Změna časování komponent	39
2.11	Příklad prezentace o dvou komponentách	40
3.1	Workspace Components	42
3.2	Workspace Presentation	45
3.3	Kontrola komponenty před jejím použitím v prezentaci	46
3.4	Workspace Slideshow	49
3.5	Workspace Presenter	49
C.1	Ukázka z workspace Presentation	80
C.2	Ukázka z workspace Presenter	81
C.3	Ukázka z workspace Slideshow	82

Seznam tabulek

1.1	Naplnění potřeb v rámci jednotlivých skupin	9
1.2	Naplnění požadavků uživatelů	28
4.1	Výsledky uživatelského testování	64

Úvod

Pro vytváření prezentací a samotné prezentování dnes existuje nespočet nástrojů. Téměř všechny se omezují na prezentace ve 2D. Tato práce představuje alternativu, která díky navázání na prostředí aplikace Blender využívá i třetí rozměr. Jsou zde popsány dílčí fáze vývoje. Jejich výsledky jsou shrnuty v závěru.

Představení problému

Prezentace se stala téměř každodenní činností pro každého z nás. Existuje mnoho nástrojů, které se této oblasti věnují. Nazývají se prezentační systémy. V některých situacích však stávající řešení nestačí.

S potřebou prezentování se v dnešní době setkal asi každý. Při prezentování je cílem srozumitelně představit posluchačům vybrané téma, sdělit myšlenku či předat informaci. A to ideálně formou, která přitáhne a udrží pozornost posluchače. Prezentace se stala téměř denní záležitostí našich studijních, pracovních i rodinných životů.

Existuje velké množství nástrojů, které odpovídá rostoucí potřebě prezentovat. Tzv. prezentační systémy jsou velice rozšířenou kategorií softwaru a nabízí uživatelům mnoho možností. Jejich hlavním cílem je prezentace nejen vytvářet, ale také prezentovat a případně ozvláštnit. Často prezentující člověk jistě ocení, když mu systém práci ulehčí nebo urychlí.

Mezi nejtypičtější zástupce patří lineární prezentační systémy, které serializují slidy za sebe. Možná ale právě nadužívání některých systémů vedlo k otupění posluchačů. Vzniká tak stále více nových nástrojů, které přinášejí prezentujícímu zajímavější přístup k prezentaci než lineárně založené systémy.

Vyšší nároky prezentujících jsou logickou reakcí na zvyšující se nároky posluchačů. Zaujmout, či dokonce překvapit, je čím dál těžší a v některých situacích tak stávající řešení nestačí. Potřeba inovovat je naštěstí v lidské po-

vaze. K zešvednělým lineárním prezentačním nástrojům dnes existují varianty, které implementují alternativní přístupy k prezentaci. Právě další takovouto alternativu představuje tato bakalářská práce.

Cíle a motivace práce

Cílem práce bylo vytvořit prezentační systém založený na aplikaci Blender. Ten měl umožnit jak tvorbu, tak prezentování myšlenek ve 3D prostředí. Toto téma jsem si zvolil, protože nebylo dosud uspokojivě vyřešeno.

Konkrétním výstupem práce je rozšíření aplikace Blender (plug-in), které umožňuje tvorbu multimediálních prezentací využívajících 3D. Prostředí Blenderu je pro takovou činnost vhodné, jelikož práce s 3D modely a jejich animace je jedním ze stavebních prvků tohoto softwaru. Kromě toho je uzpůsoben i pro přehrání videa a zvuku.

Práce je určena uživatelům, kteří hledají nový způsob prezentování. Toho je dosaženo především pomocí využití třetího rozměru. Specifikem tohoto prezentačního systému je navázání na software Blender. Zkušený uživatel Blenderu tak může s dopomocí plug-inu vytvořit animovanou 3D prezentaci. Systém však nabízí i možnost vytvoření preztační pro uživatele, kteří s 3D modely nemají žádnou zkušenost.

Téma jsem si zvolil, neboť oblast prezentací využívajících třetí rozměr je poměrně nová a zajímavá problematika, která dosud nebyla dostatečně prozkoumána. Hlavním přínosem práce je již zmíněná alternativa pro prezentující, kteří se poohlíží po nové formě představení jejich myšlenek. Přispěje k zaujetí posluchače a vytrhne ho z monotónního světa běžných prezentačních systémů.

Struktura práce

Tato práce dále pokračuje v následující struktuře: analýza, návrh, implementace, testování. Jednotlivé kapitoly jsou postupně věnovány dílčím fázím vývoje. Jejich výsledky jsou shrnuty v závěru.

V rámci analýzy současných řešení jsem se věnoval především existujícím prezentačním systémům a jejich vybraným vlastnostem. Prozkoumal jsem samotné možnosti a omezení aplikace Blender, která mi dále posloužila jako hlavní vývojové prostředí. Zanalyzoval jsem již existující rozšíření systému Blender věnující se prezentaci. Dále jsem provedl analýzu uživatelů, ze které vyplynuly typické případy užití a požadavky uživatelů.

V návrhu jsem vyšel z provedené analýzy a navrhl systém na bázi aplikace Blender. Návrh zohledňuje požadavky na vytváření samotných 3D „slidů“ (komponent), jejich spojení ve výslednou prezentaci, stejně jako samotné prezentování. Nechybí ani návrh uživatelského rozhraní. Dále je v něm naznačena

cesta pro automatizaci procesu, tedy generování jednotlivých komponent, či přímo celé prezentace. Vybrané funkcionality z návrhu byly následně zařazeny do implementace.

Do implementace samotného systému jsem zahrnul nepostradatelné funkce z návrhu. Vytvořil jsem uživatelské rozhraní plug-inu pro vytváření komponent, sestavení prezentace a samotné prezentování. Pro změny v rámci celé prezentace – např. změna pořadí slidů, či jejich přecházení – jsem využil existující UI aplikace Blender. Prezentujícímu jsem umožnil jednoduché ovládání prezentace. Nakonec jsem prozkoumal možnosti generování a automatizace. Jejich implementace ve větší míře je však mimo rozsah této práce.

Při testování jsem prozkoumal míru pokrytí případů užití. Ta se ukázala jako dostatečná a výstup implementace tak splnil vytyčené požadavky pro prezentační systém. Dále jsem provedl uživatelské testování, které odhalilo drobné implementační nedostatky a otestovalo i UI plug-inu. Identifikované problémy byly ve velké míře vyřešeny. Na ostatní chyby je uživatel buď upozorněn, instruován jak se jim vyhnout, nebo budou tématem budoucích snah.

Analýza

Analýzu jsem rozdělil do tří částí. Nejprve jsem zanalyzoval současné prezentační systémy a existující Blender plug-iny věnující se prezentaci. Dále jsem prozkoumal možnosti strukturování prezentace, nároky na obsah a vizuální stránku prezentací. Nakonec jsem vytyčil uživatelské role a jejich potřeby. Tyto potřeby byly následně pokryty pomocí případů užití.

1.1 Analýza současných řešení

V analýze současných řešení se jsem zabýval existujícími prezentačními systémy a jejich vlastnostmi. Dále pak existujícími Blender plug-iny s tematikou prezentace a samotnými možnostmi aplikace Blender. Ukázalo se, že s výjimkou webového rozhraní Emaze [1] dostupné prezentační systémy vytváření 3D prezentací neumožňují. Stejně tak jsem nenalezl rozšíření pro software Blender [2], které by využívalo třetí rozměr – vstupem zkoumaných plug-inů jsou 2D obrázky, výstupem je 2D slideshow ve formě videa.

U existujících prezentačních systémů jsem provedl rozbor vybraných vlastností. Každý má své výhody a nevýhody, žádný nástroj není všestranný a hodí se tak pro konkrétní situace více, či méně. Například aplikace PowerPoint [3] se soustředí na snadné vytváření především statických slidů a je tak často používána pro prezentace s účelem vzdělat posluchače. Oproti tomu online nástroj Powtoon [4] se zaměřuje na dynamické animace, které prezentaci ožíví a upoutají posluchačovu pozornost – je tak většinou používán k marketingovým účelům.

Analýzu současných řešení jsem dále rozdělil do 3 podkapitol. V podkapitole Existující prezentační systémy jsem provedl rozřazení do několika skupin, které jsem dále podrobněji zkoumal. V podkapitole Potřeby prezentačního systému jsem vytyčil nezbytné funkce prezentačního systému a provedl porovnání jejich naplnění napříč skupinami. Nakonec jsem v podkapitole Možnosti

aplikace Blender a jeho plug-inů zkoumal dostupné plug-iny věnující se prezentaci a samotné možnosti jmenovaného 3D softwaru.

1.1.1 Existující prezentační systémy

Identifikoval jsem čtyři skupiny prezentačních systémů podle jejich výstupu. Nejrozšířenější jsou lineární prezentační systémy, které slidy serializují. Nelineární prezentační systémy opouští myšlenku slidu a vytvářejí spíše mind mapy. Video prezentační systémy vytvářejí videozáznam prezentace. Trojrozměrné prezentační systémy využívají pro prezentaci i třetího rozměru.

V současné době existuje mnoho různých prezentačních nástrojů a každý se snaží přinést uživateli něco nového. Pro účely porovnání jsem je rozdělil do 4 kategorií podle toho, jaký formát má výsledná prezentace, tedy co je jejich výstupem:

- Lineární prezentační systémy
- Nelineární prezentační systémy
- Video prezentační systémy
- Trojrozměrné prezentační systémy

Avšak i v rámci jednotlivých kategorií nalezneme rozdíly. Liší se funkcionalitami i cenou. Existují jak aplikace, tak webové rozhraní pro tvorbu prezentace. Nejjednodušší a nejvíce rozšířené jsou prezentační systémy vytvářející tzv. slideshow, tedy sérii mnohdy statických “slidů” (obrázky a text na jednoduchém pozadí), které jsou jeden po druhém zobrazovány posluchači. V posledních letech však nejsou zdaleka jedinou alternativou a stále více lidí hledá nové cesty, jak prezentovat.

1.1.1.1 Lineární prezentační systémy

Výstupem této skupiny prezentačních systémů je jednoduchá slideshow, tedy lineární série jednotlivých slidů. Vytváření prezentace je přímočaré, jelikož slidy mají vždy konstantní vzdálenost. Přechody mezi slidy mohou být různě vizualizované, vždy však lineární. Typickým zástupcem je PowerPoint.

Hlavním faktorem v této skupině je již zmíněná linearita. Ta dává prezentujícímu možnost jednoduchého ovládní prezentace a posluchači dobrou orientaci díky konstatně přibývajícím číslu slidu. Serializace ale přináší i omezení. Z principu neumožňuje zanořování témat a jednotlivé okruhy prezentace tak lze oddělit pouze rozdílným nadpisem.

Nástroje této kategorie jsou nejpoužívanější. Již zmíněný PowerPoint [3] je v posledních letech dokonce nejpoužívanějším prezentačním systémem vůbec [5]. Především cenou se mu snaží konkurovat volně dostupný OpenOffice Impress [6], či web-based Google Slides [7]. Poněkud odlišný způsob vytváření prezentace přináší Beamer [8], který je založen na sázecím systému LaTeX. Dalšími zástupci jsou Canva [9], či SlideBean [10].

1.1.1.2 Nelineární prezentační systémy

Zástupci této skupiny nabízí podobné funkcionality jako ti z předchozí kategorie s jedním velkým rozdílem – výsledek není lineární. Výstupem je mind mapy, v rámci které se lze libovolně pohybovat v obou osách, či se neomezeně zanořovat. Zástupcem této skupiny je online nástroj Prezi.

Zmíněná nelinearita je hlavním důvodem, proč čím dál více uživatelů volí tuto variantu. Schopnost rozšiřovat prezentaci v x-ové i y-ové ose, či definování slidy uvnitř jiného slidy, je pro tuto skupinu charakteristické [11]. S nelinearitou však vzniká problém orientace posluchače. Průběh prezentace je typicky reprezentován progress barem, který však z principu nemusí přibývat lineárně.

Kromě Prezi [12] do této skupiny patří například Reveal.js [13] – framework pro vytváření HTML prezentací. Z dalších představitelů uvádím nástroj Sozi [14], který v kombinaci s aplikací Inkscape [15] dokáže ztvárnit vektorové prezentace.

1.1.1.3 Video prezentační systémy

Třetí skupina sází na výstup ve formě videa. Jedna prezentace má tak vždy stejný průběh, danou délku a rychlost. Díky povaze videa lze prezentaci přehrát na libovolném médiu. Tyto služby nabízí systémy jako Powtoon.

Nástroje z této kategorie nabízí velkou možnost personalizace, živé animace poutající pozornost a dynamické přechody. Používají se spíše pro marketingové než edukativní účely. Díky dané délce má posluchač dobrý přehled o stavu prezentace. Prezentující tak nemá možnost výklad zpomalit, ani zrychlit, pouze prezentaci explicitně zastavit. Přejít z dynamického videa na statický obrázek však nepůsobí dobře.

Nástroje tohoto typu jako je Powtoon [4], jsou většinou zpoplatněné. Stejně tak Animaker [16] nabízí několik měsíčních plánů v řádech desítek dolarů měsíčně za služby, které poskytuje. Posledním zástupcem, kterého uvádím, je systém Doodly [17], který se soustředí na kreslené animace.

1.1.1.4 Trojrozměrné prezentační systémy

Poslední kategorií jsou nástroje vytvářející 3D prezentace, respektive využívající třetí rozměr. Lineární přechody jsou nahrazeny pohybem kamery po scéně. Slidy tak nemusí mít podobu 2D plochy, ale mohou být integrované do 3D prostředí. Zástupci této kategorie jsou web-based systém Emaze, nebo aplikace pro virtuální realitu NeosVR.

Fakt, že se v této kategorii nachází pouze jeden prezentační systém, naznačuje, že se jedná o novou a nepříliš prozkoumanou oblast. Emaze [1], kromě svých několika 3D šablon, neimplementuje možnost vytvořit si vlastní trojrozměrné prostředí [18]. Slidy lze umístit do připravených 2D ploch rozmístěných po scéně. Implicitně zde není žádný prvek pro orientaci posluchače, ale stačilo by slidy lineárně očíslovat. Prvek třetího rozměru každopádně prezentaci obzvláštní a upoutá pozornost.

Oproti tomu s aplikací NeosVR [19] lze docílit maximální personalizace, nelze jí však nazvat prezentačním systémem. Nabízí vytvoření prezentace pomocí importu obrázků, které jsou serializovány do 2D slideshow a umístěny do virtuálního 3D světa - metaversa [21]. Dalšími způsoby jsou vytvoření statického informačního stánku (posteru), který může být doplněn o zvukovou stopu, či vytvoření interaktivního 3D světa. Oba tyto způsoby však vyžadují znalost tzv. vizuálního programování pomocí interního nástroje LogiX [20].

1.1.2 Potřeby prezentačního systému

Pro účely analýzy jsem vytyčil požadavky pro prezentační software. Minimální požadavky jsou nepostradatelnou sadou funkcí, které by měl implementovat každý prezentační systém. Optimální požadavky jsou standardem, na který jsou uživatelé prezentačních systémů zvyklí. Míra a způsob implementace požadavků se přesto může lišit. Porovnání v rámci jednotlivých skupin z předchozí kapitoly zachycuje tabulka níže.

Minimální potřeby prezentačního SW:

- Strukturování prezentace
- Strukturování slidu
- Návaznost slidů
- Ovládání prezentace

Jakákoliv prezentace by měla být strukturována [22]. Pro jednoduchost, řekněme, do jednotlivých slidů. I samotné slidy mívají strukturu [23]. Od nadpisů několika úrovní, přes odstavce, seznamy až po obrázky a videa. Následně

Tabulka 1.1: Naplnění potřeb v rámci jednotlivých skupin

	Struktur. prez.	Struktur. slidu	Návaznost	Ovládání	Vizualizace stavu	Nelin. pohyb	Konzistence	Přehrání zvuku	Přehrání animace
PowerPoint	slidy	šablony	lineární přechody	lineární ↔	číslo slidu	ano	šablony	ano	ano
Prezi	mind mapa	šablony	nelineární přechody	nelineární	progress bar	ano	šablony	ano	ano
Powtoon	slidy	šablony	lineární přechody	play & pause	průběh videa	omezeně	šablony	ano	ano
Emaze	slidy	šablony	omezené lineární přechody	lineární ↔	progress bar	ano	omezené šablony	ano	pouze přechody
Snu Slide show Gen.*	slidy	nelze	lineární přechody	play & pause	průběh videa	omezeně	nelze	ano	pouze přechody

je třeba zajistit návaznost slidů, tedy přechody mezi nimi (ať už jakkoli vizualizované) a správné načasování v případě, že jsou animované. V neposlední řadě musí mít prezentující možnost prezentaci ovládat [24]. Nejjednodušší a nejčastější variantou je lineární přepnutí na následující, či předešlý slide. Následuje výčet potřeb optimálních.

Optimální potřeby prezentačního SW:

- Vizualizace stavu
- Nelineární pohyb v rámci prezentace
- Vizuální konzistence “slidů”
- Přehrání animace
- Přehrání zvuku

Vizualizace stavu je důležitá především pro posluchače. Výrazně mu tím usnadňuje orientaci v kontextu celé prezentace. Naopak prezentující ocení možnost nelineárního pohybu v rámci prezentace – například skočit na určitý slide, či začít prezentaci od daného bodu. Obvyklým požadavkem je také zachování vizuální konzistence slidů a tím i celé prezentace. Pro zpestření a dokreslení se pak hodí možnost přehrání animace a zvuku.

1.1.2.1 Naplnění potřeb v rámci jednotlivých skupin

Každý prezentační systém klade důraz na jiné vlastnosti. Proto může být míra a způsob naplnění vytyčených potřeb rozdílná. Odlišnosti jsem dále rozvedl v této podkapitole. Shrnutí nabízí tabulka 1.1 se zástupci jednotlivých skupin existujících prezentačních systémů.

PowerPoint [3] (lineární prez. systém) strukturuje prezentaci na slidy. Pro strukturování jednotlivých slidů nabízí několik šablon: nadpis + seznam, nadpis + podnadpis + obrázek, apod. Slidy jsou následně serializovány a jejich návaznost i ovládání prezentace je tak lineární. Díky tomu lze stav prezentace zobrazit číslem slidu a lze přeskocit na konkrétní slide. Pro vizuální konzistenci taktéž nabízí mnoho šablon [24]. Zvládá přehrát jak zvuky, tak animace.

Prezi [12] (nelineární prez. systém) dovoluje zanořování tématů v rámci mind mapy [11]. Mezi tématy lze nelineárně přeskakovat. Strukturu témat a vizuální konzistenci zajišťují šablony. Stav prezentace je vizualizován progress barem. Přehrání zvuků a animací zde není problém.

Powtoon [4] (video prez. systém) vytváří videoprezentace rozdělené na slidy. Také zde je strukturování slidů a vizuální konzistence dosaženo šablonami. Z principu prezentaci nelze ovládat, pouze spustit a zastavit. Posluchač se orientuje pomocí indikátoru průběhu videa, které lze přetáčet pouze ručně. Samozřejmostí jsou zvuky a animace.

Emaze [1] (trojrozměrný prez. systém) je na šablonách v podstatě založen. Po připravené 3D scéně (šabloně) je rozmístěno několik placeholderů pro slidy [18]. Tato místa jsou pevně daná a tím i přechody mezi nimi, ale i jejich vizuální podoba. Ovládání je lineární, stav je zobrazen progress barem. V rámci slidu lze přehrát zvuk, animaci však ne.

Snu Slideshow Generator [25] (Blender plug-in pro prezentaci) má jako vstup hotové slidy ve formě obrázků – nelze je tak dále strukturovat ani upravovat. Slidy serializuje a vytvoří z nich videoprezentaci. Stav prezentace tak indikuje průběh videa. Zvuk lze vložit do Blender Video Sequence Editoru. Animace lze použít pouze na přechody mezi slidy.

1.1.3 Možnosti aplikace Blender a jeho plug-inů

Následující analýza ukázala, že Blender splňuje vytyčené požadavky pro tvorbu prezentace a samotné prezentování. Prostředí, které nabízí, tak může být využito pro vytváření 3D prezentací. Přesto žádný z prezentačních plug-inů, které jsem podrobil analýze, 3D prostředí nevyužívá.

1.1.3.1 Naplnění minimálních potřeb prezentačního SW

Minimální potřeby jsem vytyčil v kapitole Potřeby prezentačního SW. V této kapitole jsem zhodnotil jejich naplnění v aplikaci Blender. Míra naplnění se ukázala jako dostatečná. Blender tak dle analýzy splňuje všechny minimální potřeby pro tvorbu prezentace a lze ho tak s dopomocí plug-inu proměnit v prezentační software.

Strukturování prezentace – nejčastějším způsobem jsou dvourozměrné slidy. Ty lze vytvořit jednoduše přidáním tvaru Plane (plocha), či importováním obrázku. Slidy však nejsou jedinou možností, jak prezentaci strukturovat. Ve

3D prostoru se nabízí rozdělení na scény, či komponenty, které odpovídají jednotlivým oblastem prezentace.

Strukturování samotných scén, či slidů Blender zvládá též. Import obrázku ani videa není problém, stejně tak přidání textu. Pomocí textboxu může uživatel vytvářet odstavce, seznamy i nadpisy. Jediným estetickým nedostatkem je nepodporovaný kerning fontů. Tuto vadu by však mohl vyřešit externí script.

Načasování slidů je další nezbytnou funkcionalitou. V případě animované prezentace potřebujeme, aby se samotná animace přehrála v čase, když je daný slide právě v záběru kameru. Přecházení znamená v aplikaci Blender posunutí klíčových snímků (keyframe) v čase. Při znalosti délky trvání každého slidu jde tak o prosté přičtení vypočítaného offsetu.

Ovládání prezentace je možné zvolit různě v závislosti na struktuře prezentace. Ve všech případech ale využijeme integrovanou kameru a navázání určité akce na stisknutí dané klávesy. V případě statické lineární prezentace stačí viewport skokově nasměrovat na následující, či předchozí slide. V případě komplexnější prezentace lze využít časovou osu, kterou Blender nabízí. Stejně jako u předchozí možnosti se bude kamera pohybovat po scéně na předurčená místa, avšak využití elementu času přináší možnost prezentaci animovat, vytvářet časové smyčky apod. Bližší informace o ovládání jsem rozvedl v Návruh pluginu.

1.1.3.2 Naplnění optimálních potřeb prezentačního SW

V této kapitole jsem rozvedl míru naplnění optimálních potřeb v aplikaci Blender. Ty nejsou pro tvorbu prezentace, či samotné prezentování zásadní, ale usnadňují ho. Míra naplnění se opět ukázala jako dostatečná. Spolu s trojrozměrným prostředím, které Blender implementuje, je to vhodný kandidát pro tvorbu 3D prezentací.

Vizualizace stavu prezentace – dvěma základními alternativami jsou číslo slidu a přibývající progress bar. První variantu lze v aplikaci Blender nahradit přidáním explicitního textového pole s číslem slidu. Pro druhou variantu může uživatel využít časovou osu, nebo ji lze simulovat prodlužujícím se obdélníkem/válcem v dolní části obrazovky.

Nelineární pohyb v rámci prezentace lze simulovat pomocí funkcionality přeskočení na konkrétní frame. Pomocí toho může uživatel začít prezentaci na konkrétním slidu, či na zadaný slide skočit.

Vizuální konzistence prezentace je možné dosáhnout pomocí předpřipravených šablon – pro jednoduchost si představme nevyplněné 2D slidy. Ty mohou mít podobu .blend souborů s předdefinovanými text boxy, materiály, animacemi. Uživatel tak pouze přepíše text, vybere si vyhovující materiál a vybere, kterou animaci chce v konkrétním případě použít.

Přehrání animace jde v aplikaci Blender bez problémů. Opět nám poslouží timeline (časová osa), která je pro přehrávání jak animací, tak videa optimalizovaná. Zároveň lze Blender využít k samotnému vytváření animací a ty pak v prezentaci použít.

Přehrání zvuku je možné dvěma způsoby. Prvním je přidání zvukové stopy do video editoru, který obsahuje časovou osu korespondující s timeline pro přehrání animace. Alternativou je objekt Speaker, který uživatel umístí do scény. Ten pak funguje jako lokální zdroj zvuku (narozdíl od globálního přístupu v případě video editoru). Hlasitost je tak možné zvyšovat s klesající vzdáleností od objektu Speaker.

1.1.3.3 Existující plug-iny

Vstupem všech tří zanalyzovaných rozšíření jsou externí 2D obrázky (.jpg, .png apod.), výstupem pak 2D slideshow ve formě videa. Žádný z plug-inů tak nevyužívá třetí rozměr, který Blender nabízí.

Zanalyzované plug-iny:

- SnuSlideshowGenerator [25]
- Blender-addon-dynamic-slideshow [26]
- Modern Typography Promo – Blender Video Template Project [27]

První a druhý plug-in mají téměř identické vlastnosti, respektive blender-addon-dynamic-slideshow nabízí pouze podmnožinu vlastností plug-inu SnuSlideshowGenerator. Dále se proto budu věnovat pouze později zmíněnému. Ten nabízí editaci pořadí slidů v reálném čase, nastavení přechodů mezi slidy a určení jejich délky, či přidání vizuálních efektů v rámci jednotlivých slidů. Výsledkem je 2D video prezentace.

Stejně tak šablonu Modern Typography Promo – Blender Video Template Project lze použít k vytvoření prezentace. Jedná se o placený .blend soubor, se kterým lze vytvořit dynamické video. Obdobně jako u předchozích dvou plug-inů jsou vstupem obrázky, které zde slouží jako pozadí. Následně je možné přidat text, vybrat si z připravených přechodů a vyrenderovat výsledné video.

1.2 Datová analýza

V rámci datové analýzy jsem se zabýval možnostmi strukturování prezentace a jak se struktura liší ve 2D a 3D. Prozkoumal jsem nároky na obsah prezentací s důrazem na multimédia a 3D. Dále jsem podrobil analýze vizuální stránku prezentací a především její zachování v kontextu celé prezentace. Nakonec jsem shrnul rozdíly 2D a 3D prezentace.

1.2.1 Struktura

Nejčastějším způsobem strukturování prezentace je rozdělení na slidy. Ty jsou typicky opatřeny nadpisem a následně doplňujícím textem, či obrázky. Pro strukturování slidu se často používají šablony. Ve 3D lze rozšířit myšlenku slidu na trojrozměrné scény. Jejich strukturování je komplexnější.

1.2.1.1 Struktura prezentace

Prezentaci je třeba strukturalizovat na tématické celky. Jedno téma je typicky dále rozděleno na menší části – běžně se setkáme s pojmy jako „slide“, či „snímek“. V trojrozměrném prostředí se nabízí rozdělení na scény, po kterých se může libovolně pohybovat dynamická kamera.

Členění prezentace na menší, samostatné kapitoly je jedním ze základních požadavků při vytváření prezentace [22]. Dílčí části oddělují jednotlivá témata a přispívají tak k lepší orientaci v kontextu celé prezentace. Tzv. slidy, snímky, či stránky jsou pro většinu prezentačních systémů základním stavebním kamenem.

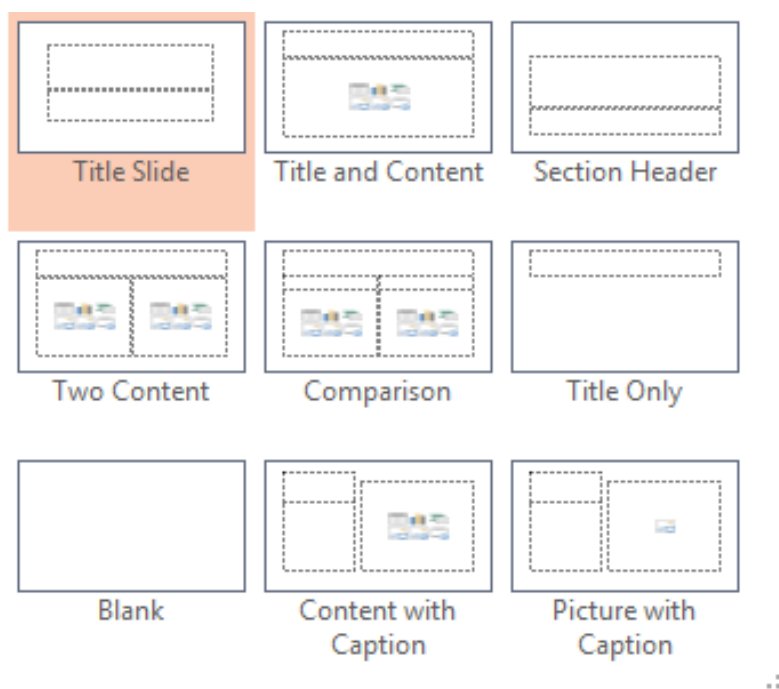
V případě lineárních prezentačních systémů jsou slidy samy o sobě nezávislé celky. Může je spojovat jejich tematika, jsou však vždy serializovány za sebe. U nelineárních prezentačních systémů může docházet k zanořování a jeden slide tak může být součástí slidu jiného. Můžeme tak uvažovat spíše o výřezu nebo oblasti zobrazení než o slidu. Prezentace pak může vypadat jako pohyb kamery po dvourozměrné ploše.

Pokud prezentaci přesuneme do 3D, je situace obdobná jako u nelineárních prezentačních systémů (Prezi) s přidáním třetího rozměru. Prezentaci dělíme spíše na scény, či komponenty, v rámci kterých lze rozmístit jednotlivé slidy. Po scénách se může kamera libovolně pohybovat, a to ve směru všech tří os. Rovněž se nekladou meze v relativní pozici jednotlivých komponent, případně slidů. Lze je umístit pod sebe, vedle sebe, za sebe, do kruhu, či je jen volně rozmístit v prostoru.

1.2.1.2 Struktura slidu

Slide se typicky skládá z pozadí a obsahu. Pro strukturování obsahu je vhodné vyjít z připravených šablon. Ty nabízí několik typických rozložení, na které jsou prezentující zvyklí. Strukturu lze následně přeuspořádat, nebo na ní dále navázat. Ve 3D může být struktura složitější, což přináší výhody, ale i jistá omezení.

Samotné slidy, či scény mívají rovněž vnitřní strukturu. Nejjednodušším dělením slidu je pozadí a obsah. Pozadí má především vizuální hodnotu, obsah



Obrázek 1.1: Šablony rozložení prezentačního systému PowerPoint

má hodnotu informační. Vizuální stránce i jednotlivým částem obsahu jsou věnovány samostatné kapitoly (viz dále).

Nejčastějším přístupem ke strukturalizaci obsahu jsou šablony rozvržení (layout). Nabízí prezentujícímu sadu rozložení, které celý slide zpřehlední a zajistí konzistenci. Šablony vychází z typických požadavků uživatelů na vytváření slidů. Příkladem může být rozdělení na nadpis a odrážkový seznam. Rozložení lze samozřejmě následně ručně přeuspořádat, nebo přidat další obsah. Na obrázku 1.1 je ukázka šablon layoutů prezentačního systému PowerPoint [3].

Trojrozměrné scény mohou být daleko komplexnější, ale myšlenku strukturalizace na pozadí a obsah lze uplatnit i zde. Využití šablon pro rozložení textu a multimédií je obdobné jako ve 2D – pokud bychom pouze umístili dvourozměrné slidy do 3D scény, tak je dokonce totožné (viz. Emaze [1]). Šablony samotných scén vzhledem k jejich možné složitosti obecně vytvořit nelze. Je zde však možnost ručního vytvoření vlastní šablony pro účely konkrétní prezentace. Tuto 3D šablonu pak stačí naplnit daty.

1.2.2 Obsah

Typickou součástí prezentace je text, který je vedle mluveného projevu hlavním nositelem informace. Ten bývá pro lepší ilustraci doplněn obrá-

zky, videi, či zvukovou stopou. 3D modely jsou nevšedním zpestřením prezentace, většina současných prezentačních systémů jejich využití nepodporuje.

1.2.2.1 Text

Text je základem, bez kterého se téměř žádný prezentační systém neobejde. Slouží jako zdroj informací doplňující mluvené slovo. Zároveň napomáhá prezentujícímu i posluchači k lepší orientaci. Text se v prezentaci objevuje ve formě nadpisů, seznamů, popisků, ale i úryvků, či částí kódu.

Klasickou prezentaci si lze bez textu jen těžko představit. Psané slovo je fundamentálním prvkem, který se v prezentacích objevuje v několika podobách. Nadpis slouží k uvedení tématu a lepší orientaci. Odrážkové seznamy shrnují myšlenky do klíčových slov, které mají význam pro prezentujícího i posluchače. Popisky a úryvky mohou být neohrazený textový objekt, nebo je lze uzavřít do text boxu.

S textem souvisí i možnost jeho formátování. Zarovnání, odsazení, nebo zalomení jsou obvyklými operacemi. U samotného písma je na místě změna fontu, barvy, velikosti. V případě text boxu může mít uživatel potřebu změnit jeho tvar a barvu. Pokud text přesuneme do 3D, lze definovat i jeho hloubku a pracovat s ním jako s 3D objektem.

1.2.2.2 Multimédia

Obrázky, grafy, video, zvuk a animace doplňují prezentaci o audiovizuální složku. To napomáhá k lepší představivosti a ilustraci předaných informací. Prezentace je rovněž díky multimédiím atraktivnější a nepůsobí stroze.

Vizuální prostředky mohou být statické (obrázky, grafy), nebo dynamické (video a zvuk, animace). Obohacují slidy a usnadňují posluchači pochopení prezentovaných myšlenek [28]. Zároveň ale přitahují pozornost a jejich nadměrné použití tak může mít i negativní efekt.

Grafické prvky jsou nedílnou součástí dobře zpracované prezentace. Správně navržený prezentační systém implementuje jejich snadné vložení a jednoduchou úpravu. Uživatelé jsou na import většiny multimédií zvyklí pomocí funkce drag-and-drop. Následná manipulace zahrnuje změnu velikosti a pozice. U dynamických multimédií je třeba vyřešit jejich načasování.

1.2.2.3 3D modely

Jen několik málo současných prezentačních systémů podporuje import a animaci 3D modelů. Jednou z výjimek je PowerPoint z balíčku Microsoft Office 2016. 3D modely jsou zajímavou formou ozvláštňení prezentace a

zároveň mohou přispět k lepší vizualizaci.

Naprostá většina dnešních prezentačních systémů se spokojí s vytvářením prezentací ve 2D. Import 3D modelů tak není obvyklou funkcí. Výjimku tvoří systém PowerPoint [3] od verze 2016 dále. Ta nabízí jak možnost importu vlastního 3D modelu, tak knihovnu volně dostupných modelů [29]. Na modely lze následně aplikovat jednoduché animace, např. rotování objektu o 360°. Je tak částečně možné simulovat trojrozměrné prostředí.

3D modely nemůžou kompletně nahradit veškeré obrázky. V některých případech však může prezentující 3D modelem dosáhnout větší přehlednosti [30]. Např. místo několika pohledů na objekt z různých úhlů může využít rotující 3D model. Posлуhač si takto objekt lépe představí. Zároveň takováto vizualizace jistě upoutá pozornost publika a ozvláštňuje celou prezentaci.

1.2.3 Vizualní podoba

Vizuální stránka slidů či scén bývá konzistentní v celé prezentaci. Dosahuje se toho často pomocí šablon. Dalším grafickým prvkem jsou přechody mezi jednotlivými částmi prezentace. Přechody mohou být statické, nebo dynamické.

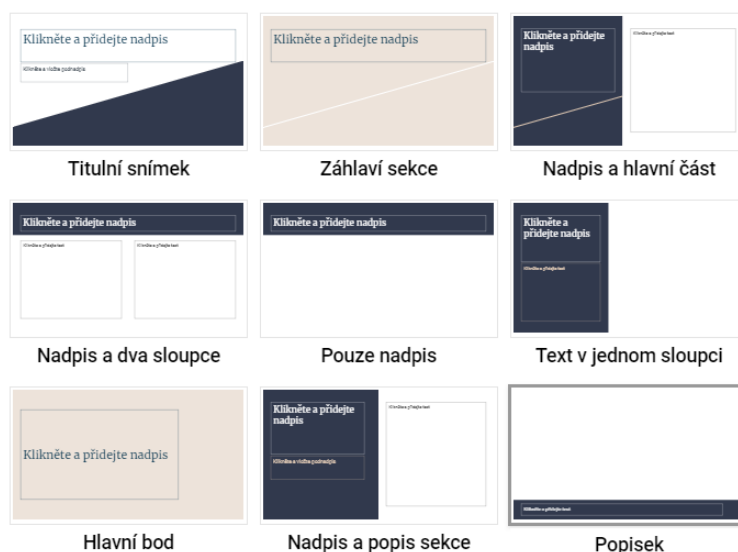
Vedle informací obsažených v prezentaci je grafická stránka druhým podstatným aspektem každé prezentace. Často rozhoduje o tom, zda si prezentující udrží pozornost posluchače, či nikoliv. Vzhledu individuálních slidů a přechodům mezi nimi je tak vhodné věnovat dostatečný čas.

K zachování vizuální soudržnosti prezentace se hodí připravené šablony. Jedná se o sadu motivů, které se starají o grafickou stránku jednotlivých slidů, především jejich pozadí. Díky šablonám jsou v prezentaci jednotné barvy, fonty, či zarovnání textu. Šablony se mohou vizuálně měnit v reakci na strukturu slidy – viz. obrázek 1.2 ze systému Google Slides [7].

Vzhledem ke struktuře prezentace je také třeba zajistit vizuální návaznost jednotlivých částí prezentace. Děje se tak pomocí přechodů. Ty mohou být čistě statické, nebo dynamické. Statickým přechodem rozumíme přeskočení na následující/předchozí slide bez jakéhokoli efektu. Příkladem dynamického přechodu je interpolace. V 3D prostředí může být přechod realizován pohybem kamery po křivce, její rotací apod.

1.2.4 Rozdíly 2D a 3D prezentace

Využití třetího rozměru přináší do prezentace nové možnosti. Zároveň je však třeba řešit problémy, které ve 2D nenastávají. Prezentace mohou být komplexnější, a jejich vytváření tak může být časově náročnější.



Obrázek 1.2: Vizuální změna šablony v reakci na její strukturu – Google Slides

Zde je shrnující výčet výhod a možností, které mohou 3D prezentace oproti těm dvourozměrným mít:

- Podrobnější strukturalizace témat do scén a slidů
- Libovolná relativní pozice jednotlivých scén a slidů
- Neomezený pohyb kamery po 3D prostředí
- Plné využití 3D modelů
- Animované scény

Trojrozměrný prostor nabízí lepší možnost strukturalizace. Každému tématu prezentace může odpovídat jedna komponenta, v rámci které lze definovat jednotlivé slidy. Ty mohou mít vůči sobě libovolnou pozici a lze je dokonce zanořovat. S tím souvisí i pohyb dynamické kamery, která může měnit svou pozici, rotaci a dokonce i velikost (poměr stran, pozorovací úhel, ...). 3D modely skutečně využívají třetí rozměr – na rozdíl od PowerPointu 2016, kde ho pouze simulují. Navíc mohou být animované, stejně jako celé scény.

S novými možnostmi přichází i nová omezení a problémy, které je potřeba řešit. Jsou to např.:

- Neohraničený prostor scény/slidu
- Načasování animací
- Nutnost ruční tvorby vlastních šablon

- Časově náročnější tvorba prezentace
- Vyšší nároky prezentace na HW

Trojrozměrné komponenty, či slidy ve 3D nejsou nijak omezeny, nemají pevné hranice. To může způsobit problémy s viditelností. Je třeba si rozmyslet, co všechno by měl a neměl posluchač v danou chvíli vidět. S tím souvisí i nutnost načasovat animace tak, aby se přehrály ve chvíli, když je daný slide na řadě. Dále může být vytváření 3D prezentací časově náročnější. Kvůli komplexnější struktuře platí rovněž nutnost vytváření vlastních šablon slidů. Nakonec při prezentování jsou kladeny vyšší nároky na výpočetní hardware, především na grafickou kartu. Dá se jim však částečně předejít, například předpočítáním stínů, vypálením textur apod.

1.3 Analýza uživatelů

V rámci analýzy uživatelů jsem se zaměřil na uživatelské role. Jeden uživatel prezentačního systému může být postupně v různých rolích a pokaždé mít jiné požadavky. Konkrétní potřeby jsem zadefinoval pro každou uživatelskou roli. Jejich naplnění se následně ukázalo v podkapitole Případy užití.

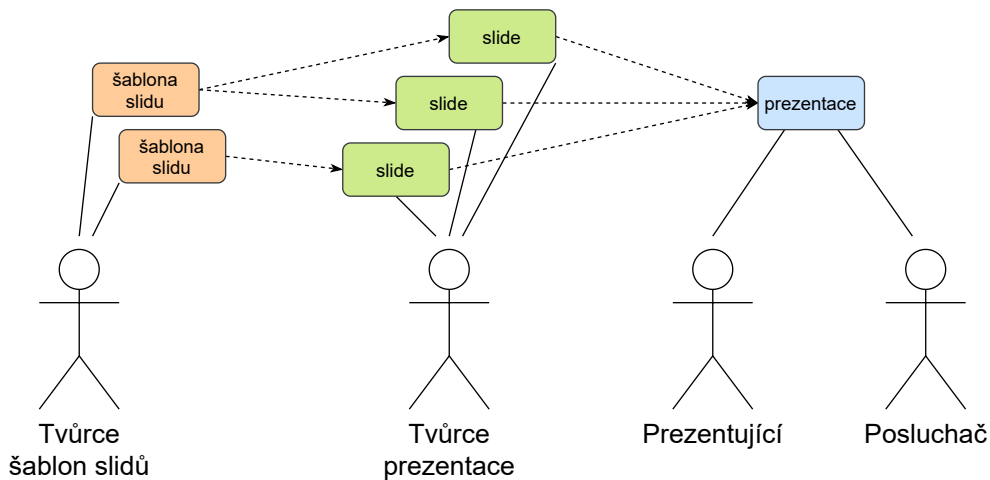
1.3.1 Uživatelské role a jejich potřeby

Vytyčil jsem celkem čtyři role, ve kterých se může uživatel výsledného prezentačního systému ocitnout – tvůrce šablon slidů, tvůrce prezentace, prezentující a posluchač. U jednotlivých rolí jsem rozvedl jejich charakteristiky a především jejich rozdílné potřeby.

Než se prezentace dostane před posluchače, je ve většině případů potřeba následujících kroků. Tyto kroky může postupně vykonávat jednotlivec, nebo se na nich může podílet více uživatelů:

1. Analýza a sběr dat
2. Grafický návrh slidů
3. Naplnění slidů daty
4. Serializace slidů
5. Odprezentování

Analýza probíhá mimo samotný prezentační systém. Nezávisle na ní může probíhat grafický návrh, během kterého vznikají „prázdné“ slidy, či šablony. Mnoho prezentačních systémů nabízí knihovnu takovýchto grafických šablon.



Obrázek 1.3: Proces vytváření prezentace

Následně se do šablon doplní konkrétní informace a multimédia. Připravené slidy jsou serializovány do výsledné prezentace. S tou předstoupí prezentující před posluchače. Situaci znázorňuje obrázek 1.3. Tento postup samotřejmě není jediný možný, například dle [31] probíhá grafický návrh až po serializaci slidů.

Při vytváření 3D prezentace v Blenderu je třeba kvůli složitější struktuře scény vytvořit šablony slidů, či scén ručně. Takto vzniklé předlohy se následně naplní daty a spojí do prezentace. Všechny tyto kroky probíhají v prostředí Blenderu a předpokládá se tak jeho uživatelská znalost. Připravenou prezentaci lze odprezentovat. Zde se již po uživateli schopnost ovládat Blender nevyžaduje.

Tvůrce šablon slidů

Uživatel v této roli navrhuje jak vizuální stránku scény (slidu), tak její strukturu. Jde tedy zejména o rovinu grafickou. Účastník vytváří tzv. šablony. Umístí tak do prostoru pouze určité placeholdery, například prázdný text box. Ty budou později doplněny o konkrétní text, obrázky atd. Návrh probíhá v prostředí Blenderu a po uživateli se tak požaduje, aby uměl tuto aplikaci obsluhovat.

F1: Vytváření slidů/scén

- Každá prezentace se skládá z menších částí. Ve 2D jsou to slidy, ve 3D scény, či komponenty. Vytváření těchto jednotlivých dílů je tak základní potřebou při tvorbě prezentace. Systém umožní jejich vytváření, úpravu a v jisté míře i generování.
- Typ: Funkční požadavek

- Priorita: Vysoká

Tvůrce prezentace

Role tvůrce prezentace zasahuje spíše do roviny informační než grafické. Tento uživatel vnese do slidů příslušná data a následně poskládá celou prezentaci. Je tedy nutné, aby si rozmyslel návaznost témat a určil jejich pořadí v prezentaci. V této roli má uživatel za úkol dostat prezentaci do finálního stavu. Pro tento proces je opět nutná znalost ovládání aplikace Blender.

F2: Vytvoření prezentace

- Když je hotový grafický návrh, je třeba naplnit prezentaci informacemi, konkrétními multimédii apod. Tento proces bude systém schopen částečně zautomatizovat. Dále je třeba komponenty uspořádat do výsledné prezentace. Systém bude umožňovat přidávání a odebrání slidů, změnu jejich pořadí, nastavení přechodů a rozmístění po trojrozměrném prostoru.
- Typ: Funkční požadavek
- Priorita: Vysoká

Prezentující

Do této role se dostane uživatel v momentu, kdy se chystá s prezentací předstoupit před publikum. Prezentaci má již ve finálním stavu a zbývá ji jen odprezentovat. Proces stále probíhá v prostředí Blenderu, ale vyžaduje se pouze minimální schopnost s touto aplikací pracovat.

F3: Prezentování

- Nachystanou prezentaci je možné odprezentovat. Prezentaci je třeba spustit, přepínat mezi jednotlivými slidy a nakonec prezentaci ukončit. Systém dá prezentujícímu možnost jednoduchého ovládání pomocí klávesnice. Měl by rovněž umožnit nelineární pohyb v rámci prezentace, např. přeskočit na vybraný slide.
- Typ: Funkční požadavek
- Priorita: Vysoká

Posluchač

V roli posluchače jsou lidé z publika. Vnímají jak mluvený projev přednášejícího, tak samotnou prezentaci. Soubor s prezentací však mohou mít stažený i ve svém zařízení. V takovém případě by mohli posluchači mít potřebu procházet

prezentaci jak současně s prezentujícím, tak nezávisle na něm.

F4: Synchronní a asynchronní pohyb v prezentaci

- Systém by mohl umožnit i přepínání mezi synchronním a asynchronním pohybem v prezentaci. Posлуhač by se tak mohl po prezentaci pohybovat společně s přednášejícím, nebo se samovolně odpojit a zobrazit si jinou část prezentace. Následně by se mohl opět synchronizovat s přednášejícím.
- Typ: Funkční požadavek
- Priorita: Nízká

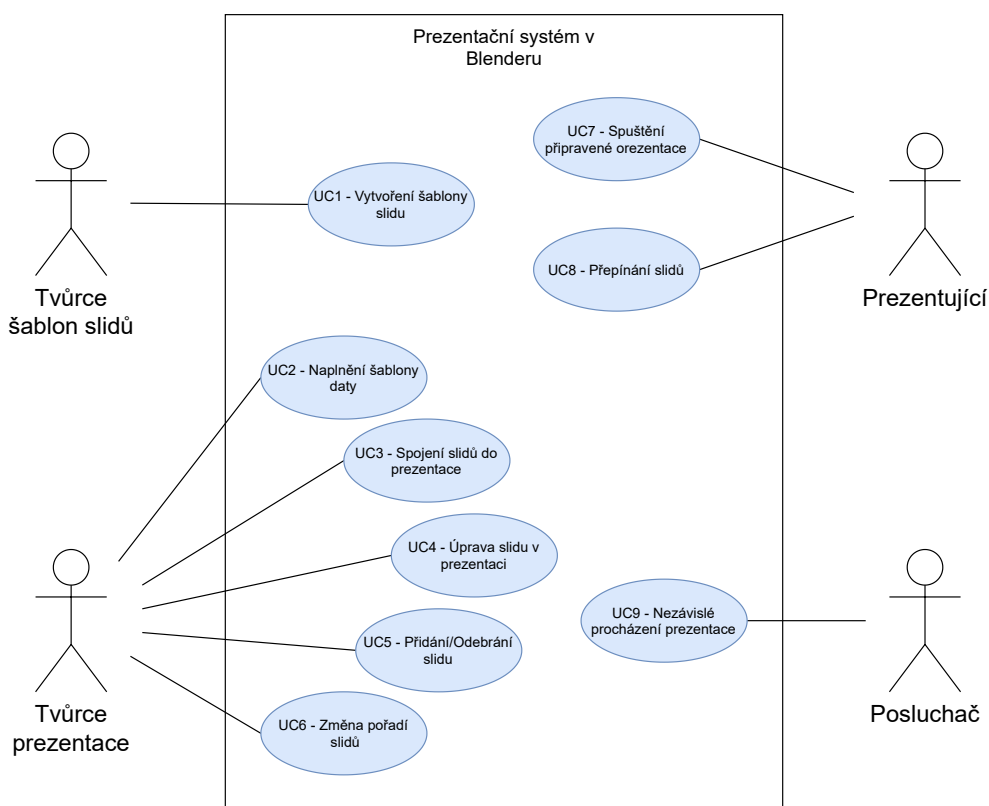
1.3.2 Případy užití

Vytvořil jsem několik případů užití výsledného prezentačního systému v Blenderu. Pokrývají potřeby uživatelů z předchozí kapitoly. Jednodušší případy jsou popsány pouze textem. Složitější případy jsou pro zvýšení přehlednosti doplněny o diagram. Nakonec jsem vše shrnul v tabulce 1.2 znázorňující konkrétní naplnění potřeb.

Tato kapitola obsahuje případy užití prezentačního systému v Blenderu. V případech užití vystupují účastníci (uživatelské role) z předchozí kapitoly. Jejich konkrétní přiřazení zachycuje obrázek 1.4. Naplnění potřeb je zobrazeno v tabulce 1.2.

UC1: Vytvoření šablony slidu

- Příklad užití umožňuje v Blenderu vytvořit novou šablonu slidu/scény. Jedná se o návrh grafické stránky slidu i jeho struktury. Takto zhotovené šablony jsou připravené, aby je tvůrce prezentace naplnil konkrétními daty. Šablony lze tvořit ručně, nebo je generovat s dopomocí plug-inu.
- Aktér: Tvůrce šablon slidů
- Hlavní scénář: Ruční vytvoření šablony pro slidy
 1. Příklad užití začíná, když se tvůrce šablon slidů rozhodne vytvořit novou šablonu pro vytváření trojrozměrných slidů.
 2. Tvůrce šablon spustí aplikaci Blender rozšířenou o plug-in umožňující vytváření šablon.
 3. Do scény si uživatel přidá libovolné objekty. Scéna se může skládat z textových objektů, 3D modelů, obrázků apod.



Obrázek 1.4: Případy užití prezentačního systému v Blenderu

4. Jednotlivým objektům přiřadí pomocí plug-inu typ z dané množiny identifikátorů, např. nadpis, obrázek apod. Všechny neoznačené objekty budou automaticky považovány za pozadí. (Tento krok není povinný. Pokud ho však uživatel přeskočí, nebude moct dále využít možnost automatického naplnění slidu daty z textového souboru - viz alternativní scénář UC2)
5. Když je s návrhem a strukturou slidu spokojen, .blend soubor uloží. V opačném případě pokračuje scénář bodem 3.

- Alternativní scénář: Vygenerování šablony pro slide pomocí plug-inu

1. Případ užití začíná, když se tvůrce šablon slidů rozhodne vygenerovat novou šablonu pro vytváření trojrozměrných slidů.
2. Tvůrce šablon spustí aplikaci Blender rozšířenou o plug-in umožňující generování šablon.
3. V UI plug-inu si vybere z nabídky typických rozložení slidu (nadpis + seznam, úvodní slide, ...).

4. Když je s volbou spokojen, stiskne tlačítko pro vygenerování a plug-in následně vytvoří příslušné objekty ve scéně. Objekty je možné dále upravit.
5. Dále scénář pokračuje bodem 3 hlavního scénáře.

UC2: Naplnění šablony daty

- Příklad užití umožňuje naplnit vytvořenou šablonu slidu konkrétními daty, případně na ni navázat. Tím vznikne komponenta připravená pro prezentaci. Konkrétní data lze doplnit ručně, či slidy naplnit s dopomocí plug-inu - v takovém případě je třeba připravit si textový soubor přesně daného formátu.
- Aktér: Tvůrce prezentace
- Předpoklady: Vytvořené šablony slidů
- Hlavní scénář: Ruční naplnění šablony
 1. Příklad užití začíná, když se tvůrce prezentace rozhodne využít připravené šablony k vytvoření komponent prezentace.
 2. Soubor se šablonou si otevře, nebo naimportuje do nového .blend souboru.
 3. Placeholdery v šabloně ručně nahradí konkrétními texty a multimédií. Případně lze ručně přidat další objekty.
 4. Když je uživatel spokojený, soubor uloží. Takto připravená komponenta je připravená pro spojení do výsledné prezentace.
- Alternativní scénář: Naplnění šablony z textového souboru
 1. Scénář začíná v 2. kroku hlavního scénáře, když se tvůrce prezentace rozhodne naplnit placeholdery v šabloně pomocí plug-inu.
 2. Uživatel vytvoří textový soubor přesně daného formátu. Každé informaci přiřadí identifikátor z dané množiny, např. nadpis, obrázek apod. U obrázku uvádí cestu k danému souboru, u textových objektů konkrétní textový řetězec.
 3. Vytvořený textový soubor vybere uživatel pomocí UI plug-inu. Plug-in se pokusí šablonu naplnit daty ze souboru.
 4. V případě dodržení správné struktury textového souboru vytvoří plug-in konkrétní komponenty, tedy naplní šablony daty a uloží je jako samostatné soubory. V opačném případě zobrazí uživateli varovnou hlášku. Po opravě textového souboru se tento krok opakuje.

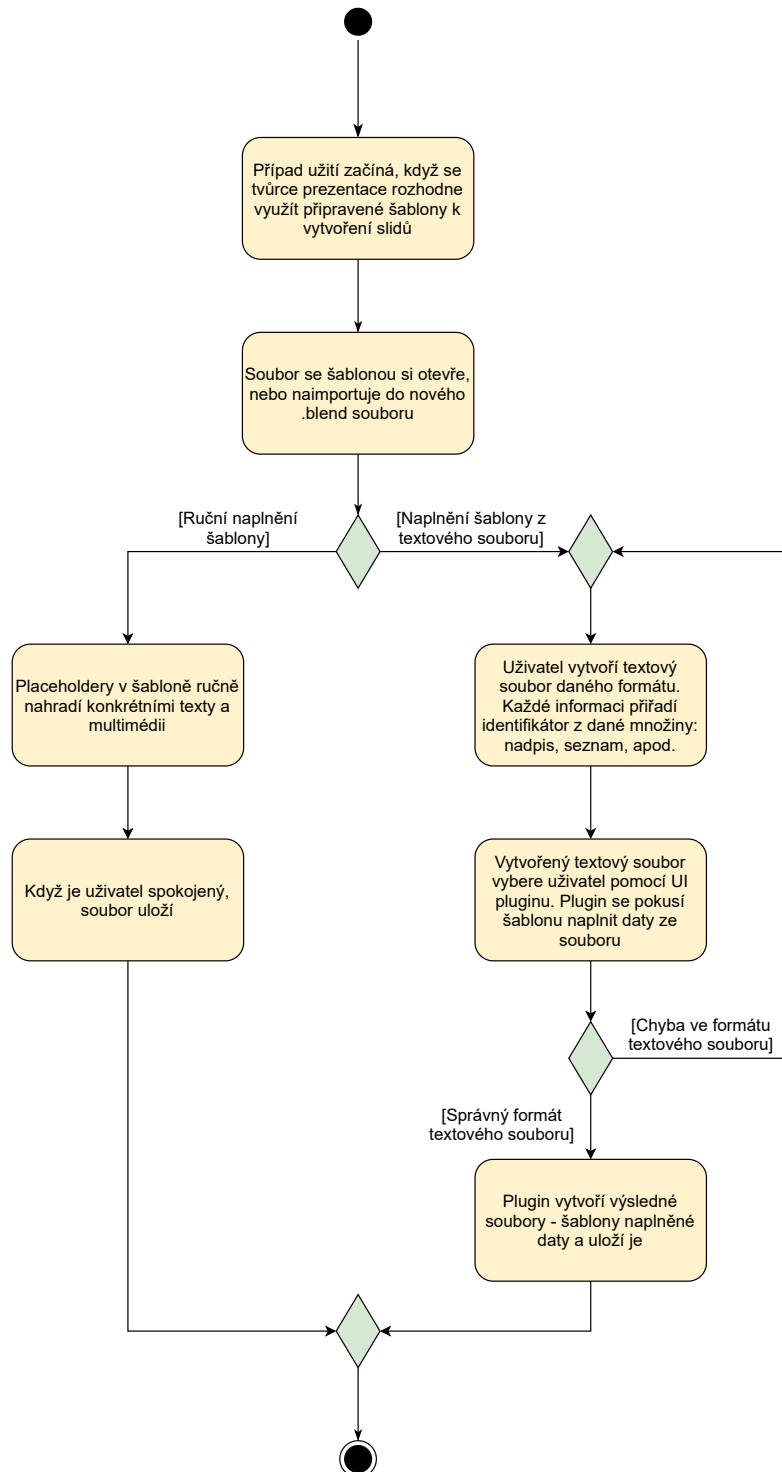
5. Plug-inem vytvořené slidy/scény jsou připraveny pro spojení do výsledné prezentace.

UC3: Spojení slidů do prezentace

- Příklad užití umožňuje spojit připravené slidy či scény do výsledné prezentace. Slidy mohou být jak animované, tak statické. Po ukončení tohoto případu užití je prezentace připravena k prezentování.
- Aktér: Tvůrce prezentace
- Předpoklady: Připravené slidy/scény pro prezentaci
- Hlavní scénář: Spojení .blend souborů do prezentace
 1. Příklad užití začíná, když se tvůrce prezentace rozhodne spojit připravené slidy ve formě samostatných souborů do prezentace.
 2. Uživatel vytvoří nový .blend soubor pro výslednou prezentaci.
 3. Pomocí UI plug-inu vybere všechny konkrétní komponenty, které chce v prezentaci použít.
 4. Uživatel si dále v UI plug-inu zvolí další možnosti prezentace, jako nastavení přechodů slidů, rozmístění slidů apod.
 5. Když uživatel vybral všechny slidy pro prezentaci a je s nastavením prezentace spokojen, zvolí v UI plug-inu možnost pro import slidů. V opačném případě pokračuje scénář bodem 3, respektive 4.
 6. Plug-in rozmístí vybrané slidy ve scéně dle preference uživatele a nastaví přechody mezi slidy, čímž je spojí do výsledné prezentace.
 7. Uživatel může následně rozmístění změnit.

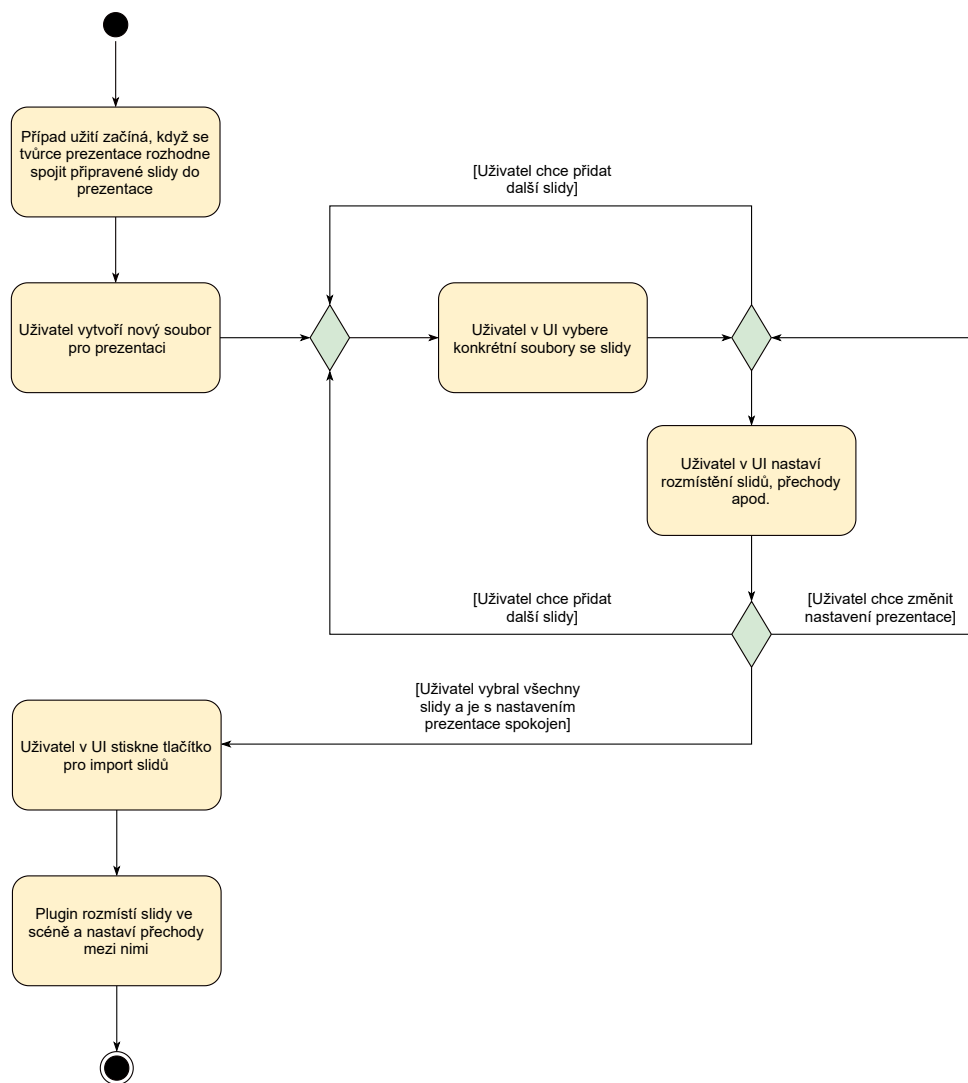
UC4: Úprava slidu v prezentaci

- Příklad užití umožňuje upravit slide, který je již součástí prezentace.
- Aktér: Tvůrce prezentace
- Předpoklady: Originální soubor se slidem
- Hlavní scénář: Úprava slidu
 1. Příklad užití začíná, když se tvůrce prezentace rozhodne upravit jeden ze slidů, který je již součástí prezentace. Předpokládáme, že má k dispozici originální soubor slidu.
 2. Uživatel otevře soubor slidu, který chce upravit.



Obrázek 1.5: UC2 – Naplnění šablony daty

1. ANALÝZA



Obrázek 1.6: UC3 – Spojení slidů do prezentace

3. Uživatel provede změny a soubor uloží. Změny se automaticky projeví ve všech souborech s prezentací, které obsahují daný slide.

UC5: Přidání/odebrání slidu

- Případ užití umožňuje přidat do prezentace další slide, případně vybraný slide z prezentace odstranit. Obě akce může uživatel provést z UI plug-inu. Nové slidy jsou implicitně přidávány na konec prezentace. Přidání i odebrání slidu může způsobit změnu rozmístění slidů - např. z n -úhelníků se stane $(n+1)$ -úhelník, respektive $(n-1)$ -úhelník.
- Aktér: Tvůrce prezentace
- Předpoklady: Prezentace s importovanými slidy

UC6: Změna pořadí slidů

- Případ užití umožňuje přeuspořádat pořadí slidů v prezentaci. U statické prezentace jde pouze o výměnu pozic dvou a více slidů - v takovém případě využije příslušná tlačítka v UI plug-inu. U dynamické prezentace, která vyžaduje korektní časování, jde o posun veškerých zasažených animací - v takovém případě uživatel využije k tomu určené UI s časovou osu.
- Aktér: Tvůrce prezentace
- Předpoklady: Prezentace s importovanými slidy

UC7: Spuštění připravené prezentace

- Případ užití umožňuje prezentujícímu spustit připravenou prezentaci a začít tak prezentovat. Uživatel prezentaci spustí v souboru s připravenou prezentací přes UI plug-inu. Plug-in se postará o další náležitosti jako režim celé obrazovky, skrytí navigačních prvků apod.
- Aktér: Prezentující
- Předpoklady: Připravený soubor s prezentací

UC8: Přepínání slidů

- Případ užití umožňuje prezentujícímu přepínat mezi jednotlivými slidy pomocí klávesnice. Vedle možnosti sekvenčního pohybu na následující, či předchozí slide, je uživateli umožněno i přeskočení na libovolný, jím vybraný slide.

1. ANALÝZA

- Aktér: Prezentující
- Předpoklady: Připravený soubor s prezentací

UC9: Nezávislé procházení prezentace

- Příklad užití umožňuje posluchači odpojit se od slidy, který je právě prezentován a pohybovat se v prezentaci nezávisle na prezentujícím - asynchronně. Může se tak vrátit ke slidům předchozím, nebo nahlédnout na slidy nadcházející. Následně se může opět synchronizovat s výkladem a sledovat tak slidy, které vidí i prezentující.
- Aktér: Posluchač
- Předpoklady: Prezentující sdílí informace o aktuálním stavu prezentace

Tabulka 1.2: Naplnění požadavků uživatelů

	Případy užití								
Požadavky	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9
F1	X	X	X						
F2				X	X	X			
F3							X	X	
F4									X

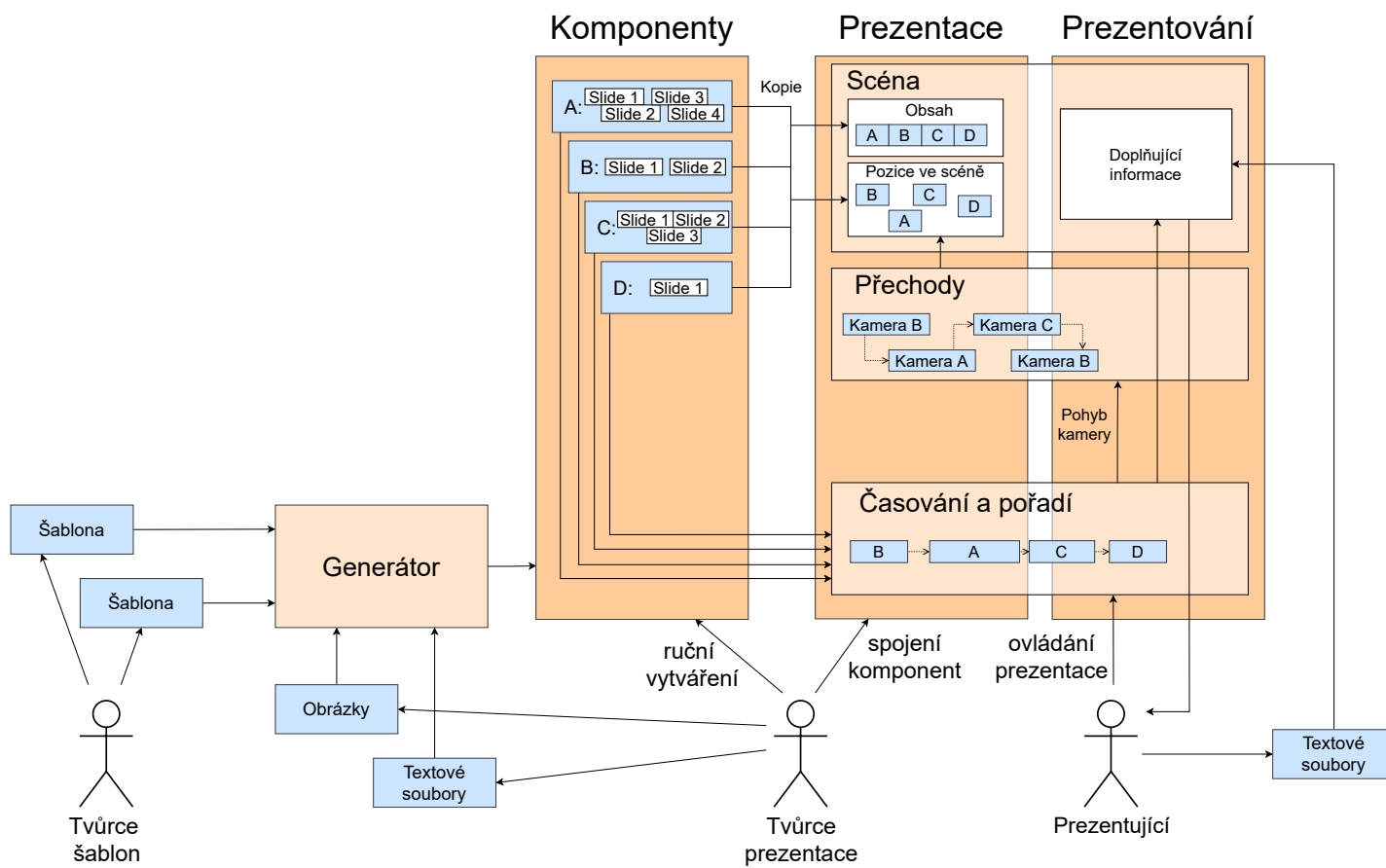
Návrh

Prezentační systém v Blenderu jsem navrhl s ohledem na výsledky předešlé analýzy. Vznikla tak rovina vytváření komponent pro prezentaci, rovina vytváření prezentace (skládání komponent) a rovina samotného prezentování. Rovina (a)synchronního pohybu posluchače po prezentaci je mimo rozsah této práce a v návrhu tak nefiguruje. Pro zbylé tři části jsem navrhl možnost jejich realizace, uživatelské rozhraní a částečně i automatizaci procesů, které s nimi souvisejí.

Návrh pracuje s uživatelskými rolemi, které vyšly z přecházející analýzy. Tvůrce prezentace tak vytvoří šablony, které mohou být použity k vygenerování konkrétních částí prezentace (komponent). Ke generování jsou dále třeba textové soubory s daty a obrázky, které budou do šablony doplněny – ty obstará tvůrce prezentace. Komponenty mohou být vytvořeny i ručně.

Celý systém je úzce propojen s časovou osou. Pro prezentaci je tak důležité znát časování a pořadí komponent, které se použije při prezentování. V závislosti na čase bude určena aktivní kamera. Od toho se odvine její pozice, přechody a viewport – tím se určí právě viditelná komponenta a její slide.

Prezentování pak znamená posun po časové ose jedním, nebo druhým směrem. Na čase závisí i konkrétní soubor s poznámkami, který se zobrazuje prezentujícímu. Celý návrh ilustruje obrázek 2.1.



Obrázek 2.1: Návrh prezentačního systému v Blenderu

2.1 Zavedení pojmů

Aby nedocházelo k omylům, objasnil jsem v této kapitole pojmy, které budou v návrhu dále často používány. Jde především o obecné pojmy spojené s prezentací a pojmy vyskytující se v Blenderu. Ukázalo se, že některá označení se překrývají a je třeba je rozlišovat. Následně jsem proto provedl jejich vzájemné porovnání a namapování.

„Scene“ (scéna), „collection“ (kolekce), či „viewport“ (pohled kamery). Pojmy, které uživatelé Blenderu bezpečně znají. Při vytváření prezentace se pak setkáme s pojmy jako slide, snímek, přechod, nebo samotný výraz prezentace. Součástí této kapitoly je zavedení a vysvětlení pojmů, které budou v práci dále používány. Zároveň jsou zde popsány jejich ekvivalenty v jazyce aplikace Blender.

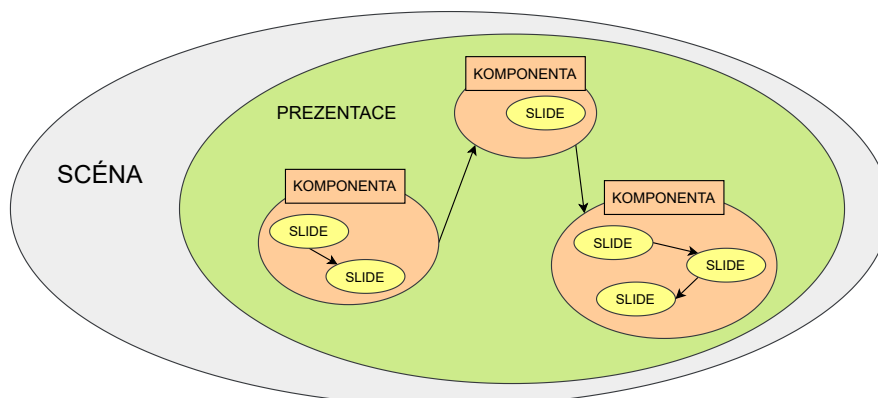
Celá 3D prezentace se nachází v trojrozměrném prostoru jedné Blender scény. Dílčí tématické části 3D prezentace proto již nelze nazvat scénami. Dále jsou tedy popisovány jako *komponenty* prezentace. Komponentu lze dále rozdělit na menší celky – určité body zájmu. Ty nazveme *slidy* a jedná se v podstatě o ekvivalent pojmu viewport. „Frame“, čili snímek, je již v Blenderu zaveden a opět ho tedy nelze použít. *Přechody* jsou pak pohyby kamery celými komponentami, případně mezi jednotlivými slidy. Vše je pro přehlednost dále shrnuto a doplněno obrázkem 2.2.

- **Slide** = Místa zájmu v rámci jedné komponenty, konkrétní výhled kamery
- **Komponenta** = Dílčí tématický celek prezentace, kolekce objektů, serializace slidů
- **Prezentace** = Spojení a serializace komponent v jedné scéně
- **Scéna** = Trojrozměrný prostor Blenderu, který obsahuje prezentaci
- **Přechod** = Pohyb kamery mezi slidy nebo komponentami

2.2 Komponenty prezentace

Pro jednotlivé komponenty prezentace jsem navrhl jejich ruční vytváření a částečně i jejich generování. Dále bylo nutné vyřešit jejich reprezentaci, vnitřní strukturu a časování. Návrh umožňuje použití jak statických, tak dynamických komponent. Pro vytváření komponent jsem navrhl samostatné uživatelské rozhraní a Blender layout.

Komponenty jsou vytvářeny a následně i spojovány v prostředí Blenderu. Skládají se z kamery, 3D modelů, obrázků, textů apod. Výsledná komponenta



Obrázek 2.2: Zavedené pojmy pro prezentaci ve 3D

musí být ve formátu, který dokáže Blender uložit a následně načíst. Nabízí se přímo nativní Blender soubor (.blend), ve kterém lze snadno definovat kameru i animace. Konkrétní volba formátu a její důvody jsou blíže popsány v implementaci.

V rámci jedné komponenty může být více slidů. Každý slide má přidělený určitý časový úsek, ve kterém bude vidět definovanou kamerou. Pokud časový úsek odpovídá jednomu snímku (jeden frame), bude kamera stát na místě a slide bude statický. Pokud je časový úsek delší než jeden snímek, může se kamera a nebo jiné objekty komponenty pohybovat.

Návrh počítá i s možností cyklického pohybu. Nastavení takového pohybu, nebo zastavení kamery připraví sám uživatel. Systém následně časování v prezentaci rozpozná, posune v čase a zareaguje dle preferencí uživatele. Toto přecházení může učinit i sám uživatel a realizovat tím změnu pořadí komponent v prezentaci. Nemělo by však dojít k překrytí jednotlivých časových úseků komponent.

Komponenty může uživatel vytvářet kompletně ručně, nebo s dopomocí plug-inu. Systém ulehčí vytváření komponent několika způsoby:

1. Generování komponent z připravených obrázků
 - Uživatel má slidy ve formě obrázků. V UI pouze zvolí relativní pozici slidů a typ kamery. Plug-in je rozmístí v prostoru a nadefinuje kameru.
2. Vytváření šablon pro generování komponent
 - Uživatel vytvoří z pomoci plug-inu šablonu komponenty. Pomocí UI přiřadí prázdným textovým objektům identifikátory. Tyto placeholdery budou později naplněny daty z externího textového souboru. Strukturu toho textového souboru vygeneruje plug-in za uživatele.

The image shows a UI design for 'Component Creation' divided into three sections:

- Image Slides:** Contains a 'Select Images:' field with a 'path' input and a file selection icon. Below it is a 'Slide position:' field with a 'position in scene' input. There is a checked checkbox for 'Interpolate camera:' and a 'Transition time:' field with a 'seconds' input. A 'GENERATE COMPONENT' button is at the bottom.
- Template Creation:** Contains an 'Object Type:' field with a 'type enum' input. Below it are two buttons: 'ASSIGN TYPE' and 'GENERATE TEXTFILE'.
- Generate from Template:** Contains a 'Textfile:' field with a 'path' input and a file selection icon. Below it is a 'Slide Template file:' field with a 'path' input and a file selection icon. A 'GENERATE COMPONENT' button is at the bottom.

Obrázek 2.3: Návrh UI pro vytváření a generování komponent

Typická rozpoložení šablon, jako nadpis + seznam + obrázek, vytvoří plug-in za uživatele.

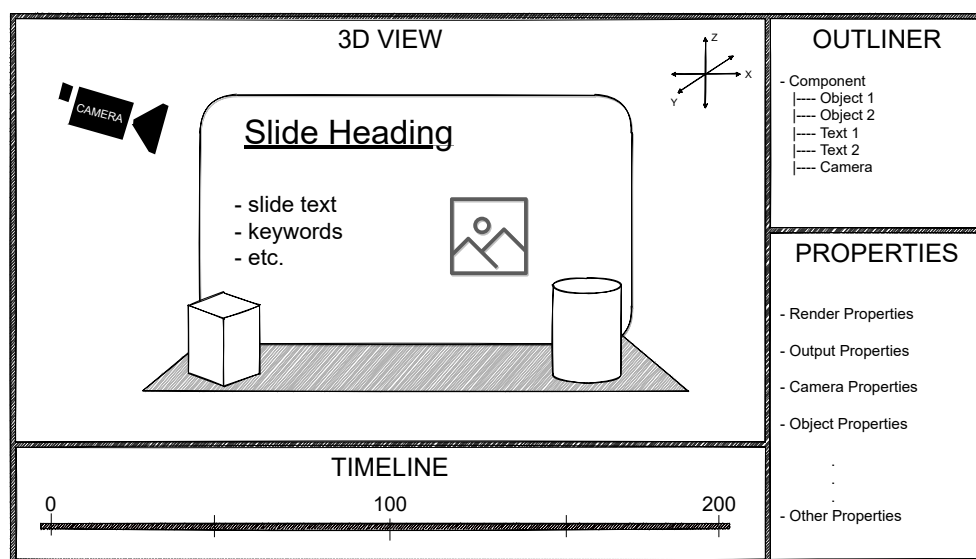
3. Generování komponent z připravených šablon

- Uživatel využije připravenou šablonu a vyplněný textový soubor pro vygenerování komponent. Placeholder objekty v šabloně jsou tak nahrazeny konkrétními daty z externího textového souboru.

Pro tyto operace jsem navrhl uživatelské rozhraní (obrázek 2.3). Dále jsem navrhl layout pro vytváření a generování komponent (obrázek 2.4). Okno „3D View“ je zásadní pro tvorbu trojrozměrného prostředí. „Timeline“, tedy časovou osu, pak uživatel ocení především při vytváření a časování animací.

2.3 Prezentace

Při tvorbě prezentace je zásadní výběr komponent, ze kterých se bude skládat. Návrh počítá s přidáváním a odebíráním komponent, či se změnou jejich pořadí. Tyto operace ve velké míře odděluje od vytváření jed-



Obrázek 2.4: Návrh layoutu pro vytváření a generování komponent

notlivých komponent. Navržený systém dále zohledňuje rozmístění komponent ve 3D prostoru a nastavení přechodů mezi nimi. Pro vytváření prezentace jsem navrhl samostatné uživatelské rozhraní a vlastní Blender layout.

Prezentace vznikne spojením komponent, které mají jednotný typ souboru. Uživatel pomocí UI plug-inu vybere příslušné komponenty. Systém nejprve vytvoří jejich kopie a poté je rozmístí v prostoru scény dle preference. Zároveň dochází k jejich serializaci – kopie komponent se přecházejí tak, aby na sebe navazovaly a nastaví se přechody mezi nimi. Tím je prezentace připravená k prezentování. Menší úpravy komponent, respektive jejich kopií, umožní systém i v kontextu prezentace. Větší úpravy však lze provést pouze ve zdrojovém souboru dané kopie komponenty.

V návrhu jsou dále úpravy takto vytvořené prezentace. V uživatelském rozhraní plug-inu lze přidat další komponenty, nebo existující komponenty z prezentace odebrat. To způsobí změnu časování a přechodů. Pro změnu pořadí komponent (jinými slovy přecházení komponent) lze využít existující UI Blenderu. Pro změnu časování jich existuje hned několik – Dope Sheet, Timeline, Graph Editor, nebo Nonlinear Animation. Konkrétní volba a její důvody jsou popsány v implementaci.

Pro nastavení prezentace jsem navrhl samostatné uživatelské rozhraní (obrázek 2.5). Pomocí něj uživatel vybere příslušné komponenty, nastaví typ přechodů mezi nimi apod. Rovněž jsem vytvořil návrh vlastního layoutu, který vytváření prezentace ulehčí a zpřehlední (obrázek 2.6). V okně 3D View má uživatel přehled o relativní pozici komponent. Camera View je výsledný po-

Presentation Setup

Import Components

Components:

Component position:

Interpolate camera:

Transition time:

CREATE PRESENTATION

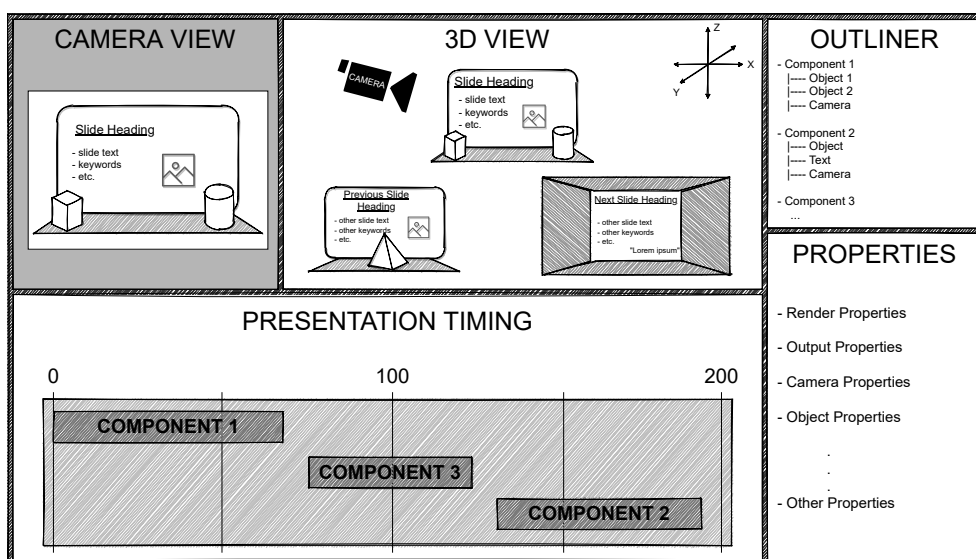
Component Settings

Selected Component:

EDIT COMPONENT

DELETE COMPONENT

Obrázek 2.5: Návrh UI pro skládání a úpravy prezentace



Obrázek 2.6: Návrh layoutu pro skládání a úpravy prezentace

hled kamerou. Presentation Timing pak využije pro posun komponent v čase a změnu jejich pořadí.

2.4 Prezentování

Prezentující by měl mít možnost prezentaci ovládat. V návrhu je tak možnost ovládání pomocí klávesnice i myši. Pohyb po prezentaci bude převážně sekvenční. Pro prezentování jsem navrhl samostatné uživatelské rozhraní. Návrh počítá i s prezentováním na více zobrazovacích zařízeních a nabízí tak dvě různá rozložení obrazovky – jedno optimalizované pro posluchače, druhé pro prezentujícího.

Ovládání prezentace je jedním ze základních požadavků při prezentování. Systém je proto navržený tak, aby ovládání umožnil pomocí klávesnice. Výchozí nastavení kláves si bude dle návrhu možné změnit podle potřeby. Dále návrh umožňuje prezentaci ovládat pomocí myši a tlačítek v UI plug-inu. Ty dovolí sekvenční pohyb v čase dopředu i zpět, nebo přeskočení na začátek, či konec prezentace.

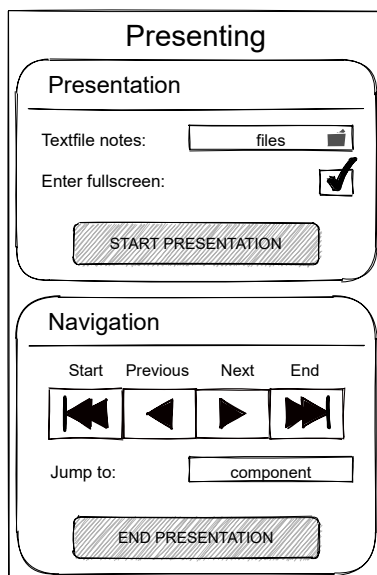
Pohyb po prezentaci bude realizován posunem po časové ose. Díky předešlé serializaci bude časem vždy určena jedna konkrétní komponenta. Tím bude také přesně určena pozice kamery, kterou se bude hledět do scény. Časové úseky mezi komponentami budou vyplněny přechody. Přechod bude realizován interpolací kamery. Skokový přechod je speciální případ, kde interpolace trvá jeden snímek a je tak nepostřehnutelná.

Kromě ovládání je v návrhu UI pro prezentování samotné spuštění prezentace (Obázek 2.7). Spuštěním se systém přepne do layoutu pro prezentování, který je navržený pro maximální zážitek posluchače (obrázek 2.8). Jsou v něm skryty všechny navigační prvky Blenderu, které by mohly při prezentování rušit. Návrh zahrnuje ještě další layout, který je určený prezentujícímu (obrázek 2.9). Díky němu má uživatel přehled o čase a může si zobrazit doplňující informace k prezentaci – poznámky ke konkrétní části, či předchozí, současnou i následující komponentu prezentace. Poznámky bude možné načíst z externího textového souboru.

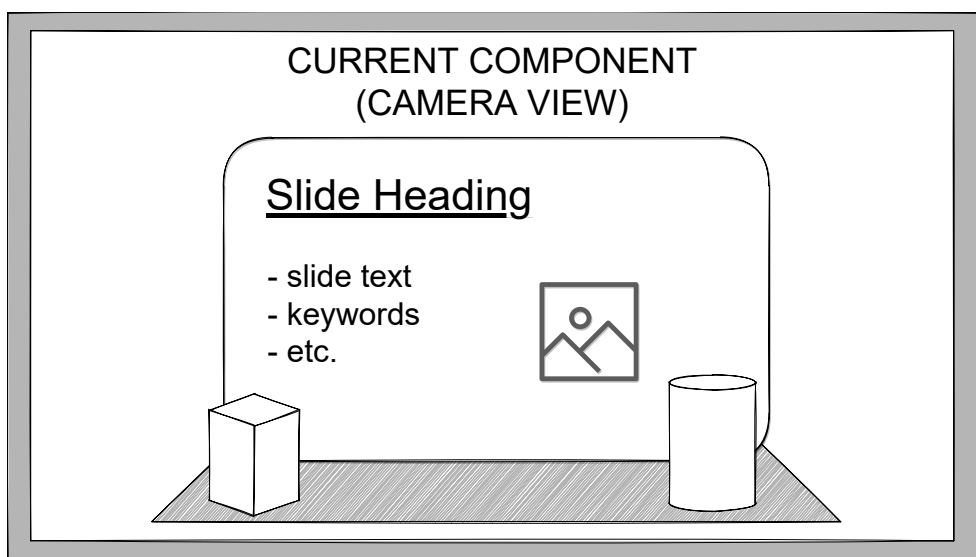
2.5 Datový a stavový model

Specifikem výsledného systému bude navázání na časovou osu. Při prezentování bude nutné znát aktuální polohu v čase prezentaci, stejně tak začátek a konec prezentace. Od času se bude odvíjet umístění aktuální kamery. Při skládání prezentace a nebo vytváření komponent bude třeba znát umístění importovaných souborů. V rámci komponent bude muset systém uchovávat relativní pozice na časové ose, kde se má budoucí prezentace zastavit, nebo zacyklit.

Uvnitř komponent je kromě obsahu třeba definovat jednotlivé slidy. Každý slide je vymezen konečným časovým úsekem. Součtem časů slidů dostaneme

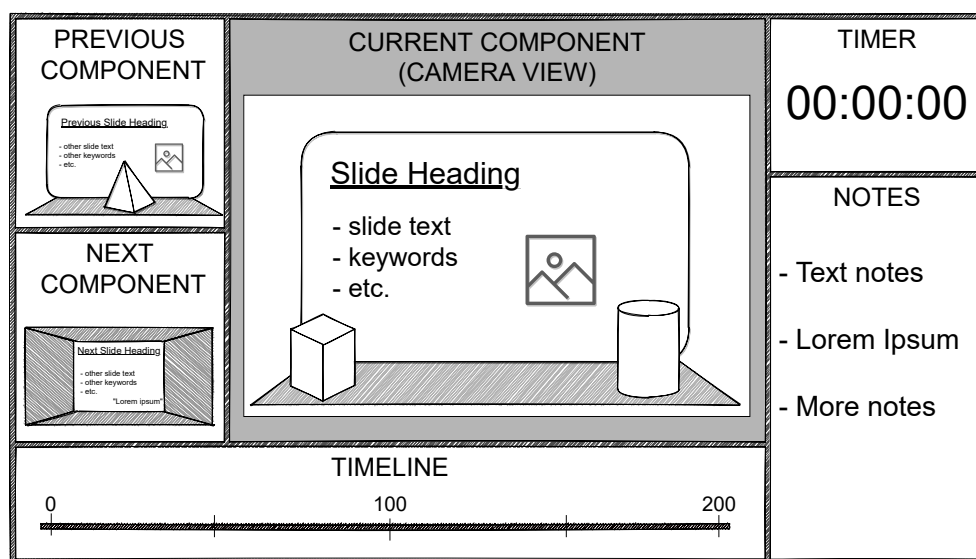


Obrázek 2.7: Návrh UI pro prezentování



Obrázek 2.8: Návrh layoutu pro posluchače

2. NÁVRH



Obrázek 2.9: Návrh layoutu pro prezentujícího

celkovou délku komponenty. Pokud čas dospěje ke konci slidu, mohou nastat dvě možnosti:

1. Čas se zastaví – tím se zastaví prezentace i veškeré její animace a obraz je tak statický
2. Čas se zacyklí – přeskočí zpět na začátek současného slidu a běží znovu ke konci slidu

Tuto volbu si musí systém uložit a ve výsledné prezentaci ji použít. U cyklických slidů si dále musí pamatovat začáteční a koncový čas smyčky.

Při vytváření prezentace musí systém vložit do scény vybrané komponenty, které jsou umístěné v souborech na disku. Uchovává si pozici ve scéně a časování každé komponenty. Díky tomu může mezi nimi přesně vytvořit přechody kamery. Zároveň vidí i na časování uvnitř komponenty, které je oproti původnímu souboru posunuté. Po poskládání prezentace zná systém celkový počet komponent a celkový čas prezentace. Ten získá sečtením časů komponent a přechodů. Změnu časování znázorňuje obrázek 2.10.

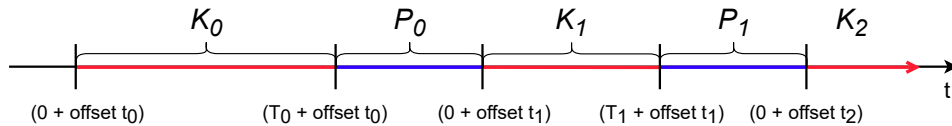
V průběhu prezentování systém čte data týkající se prezentace. Konkrétní informace jsou určeny časovou osou. Systém ví, kde dílčí komponenty a jejich slidy začínají a končí. Systém si pamatuje pořadí komponent, tudíž není problém určit tu následující, či předchozí. Situaci lze přirovnat k deterministickému konečnému automatu $M = (S, \Sigma, \delta, s_0, F)$:

- Slidy představují jednotlivé stavy: $S = \{ s_0, s_1, s_2, s_3, \dots, s_n \}$, kde $n - 1$ je celkový počet slidů všech komponent

offset $t_n = \sum_{i=0}^n t_i + t_{p_i}$, kde t_i je čas komponenty K_i a t_{p_i} je čas přechodu mezi K_i a K_{i+1}

Komponenta K_n : $t_n = \langle 0, T_n \rangle \xrightarrow{\text{přechodování}} \langle 0 + \text{offset } t_n, T_n + \text{offset } t_n \rangle$

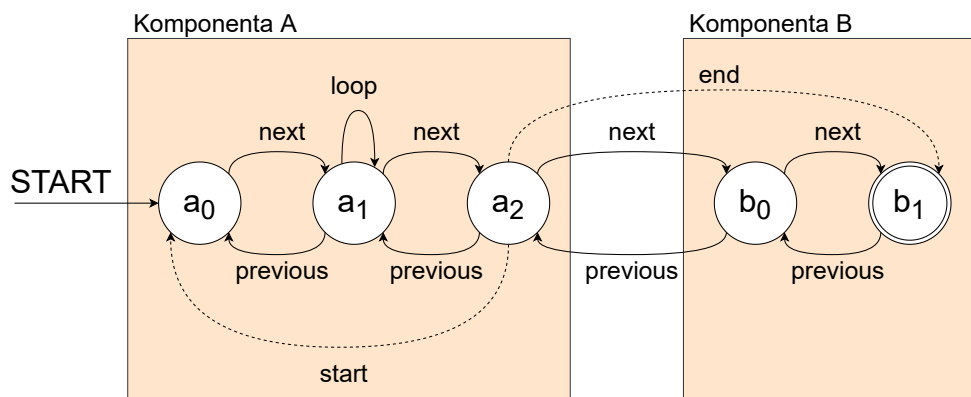
Přechod P_n : $t_{p_n} = \langle T_n + \text{offset } t_n, 0 + \text{offset } t_{n+1} \rangle$



Obrázek 2.10: Změna časování komponent

- Vstupem je ovládání prezentace, nebo indikátor zacyklení slidu: $\Sigma = \{next, previous, start, end, loop\}$
- Přechod může být na další, či následující slide v sérii. Zároveň lze z kteréhokoli slidu skočit na začátek či konec prezentace: $\delta(s_0, next) = s_1$, $\delta(s_1, previous) = s_0$, $\delta(s_1, end) = s_n$, $\delta(s_2, start) = s_0$, $\delta(s_2, loop) = s_2$, $\delta(s_2, next) = s_3, \dots$
- Počáteční stav je první slide první komponenty prezentace: s_0
- Koncový stav je pouze jeden, je to poslední slide poslední komponenty: $F = \{s_n\}$

Obrázek 2.11 není formálním znázorněním deterministického konečného automatu, ale pouhou zjednodušenou vizualizací. Ukazuje příklad prezentace o dvou komponentách A a B – první má tři slidy (a_0, a_1, a_2), druhá dva (b_0, b_1). Na druhém slidu první komponenty (a_1) je zacyklení. Přechody *start* a *end* by měly vést z každého stavu do počátečního, respektive koncového stavu automatu. Pro větší přehlednost obrázku jsou však tyto přechody naznačeny pouze u slidu a_2 .



Obrázek 2.11: Příklad prezentace o dvou komponentách

Implementace

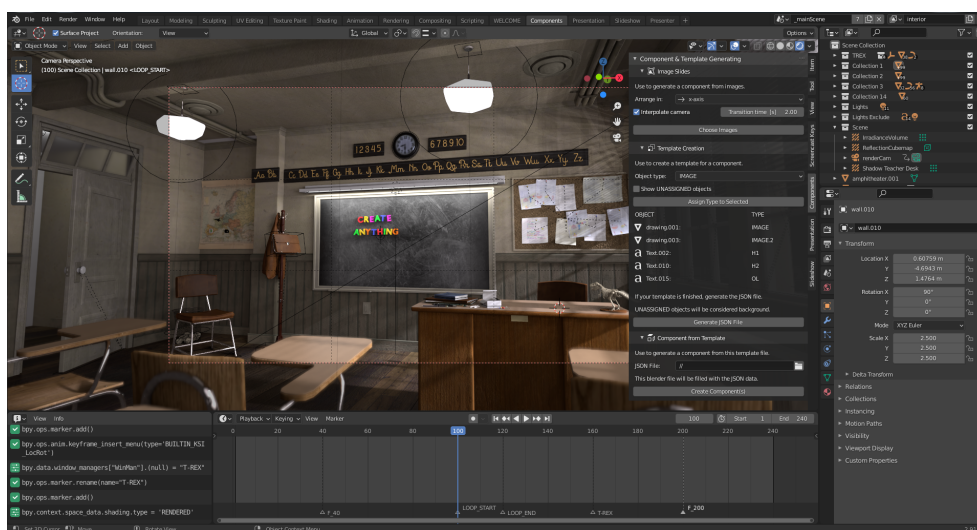
Při implementaci jsem vyšel z návrhu a vytvořil výsledný plug-in. Ten proměnil Blender v prezentační systém. Dovoluje vytváření komponent ve formátu .blend a jejich jednoduché generování pomocí šablon a obrázků. Komponenty je možné následně spojit do prezentace a díky NLA Editoru měnit jejich pořadí. Prezentující má možnost prezentaci spustit a ovládat, to vše na více obrazovkách. Dále jsou v této kapitole popsány problémy, na které jsem při implementaci narazil a možnosti budoucího rozšíření.

V návrhu jsem vytvořil koncepty uživatelských rozhraní a rozložení obrazovky (layout) pro jednotlivé roviny prezentačního systému - vytváření komponent, vytváření prezentace a prezentování. V implementaci tyto představy nabraly konkrétní podoby. Každý layout byl realizován pomocí tzv. Blender „workspace“. Workspace je přednastavené rozložení oken, Blender jich již několik obsahuje – Modeling, Shading, Scripting a další. Jejich vytváření pomocí skriptu je však zdlouhavé. Proto jsem přišel s alternativním řešením.

Workspaces jsem vytvořil ručně a uložil je do souboru PRESENTATION.TEMPLATE.blend. Tento soubor tak slouží jako výchozí šablona a obsahuje navíc workspace Component, Presentation, Slideshow a Presenter. Ty vychází z návrhu layoutů a jsou optimalizovány pro jednotlivé operace. Aby je uživatel mohl využít i ostatních .blend souborech, je potřeba v souboru šablony zakliknout *File – Defaults – Save Startip File*. Tím se workspaces uloží do UI Blenderu.

S jednotlivými workspaces korespondují 3 uživatelská rozhraní – opět Component, Presentation, Slideshow. Ty vychází z návrhu a zahrnují funkcionality odpovídající jednotlivým rovinám prezentačního systému.

3. IMPLEMENTACE



Obrázek 3.1: Workspace Components

3.1 Komponenty

Komponenty mají jednotný typ souboru, konkrétně .blend. Veškeré objekty dané komponenty jsou součástí jedné kolekce, která se poté stane součástí prezentace. Jednotlivé slidy jsou definovány pomocí markerů. Název markeru určuje, zda se má čas zastavit, či zacyklit. Komponenty lze vytvářet ručně, pomocí šablon, nebo z obrázků. Pro jejich vytvoření se hodí workspace Components 3.1.

Při volbě, jaký typ souboru zvolit pro komponenty, padla volba na nativní soubor Blenderu, tedy .blend. Ten v sobě uchovává jak stromovou strukturu objektů ve scéně, tak jejich animace, kameru, materiály, metadata apod. Zároveň ho lze lehce vložit do jiného .blend souboru – v našem případě do souboru s prezentací. První možností je link souboru, druhou append. Obě možnosti jsou blíže popsány dále v podkapitole Prezentace.

V souboru s komponentou lze mít i další objekty, které v prezentaci uživatel použít nechce, například referenční obrázky. Proto je zde možnost tyto objekty odlišit pomocí kolekcí. Objekty komponenty by měli být v kolekci „Component“. Pojmenování ostatních kolekcí může být libovolné, systém je bude každopádně ignorovat. Zároveň platí, že v jedné komponentě, respektive kolekci, lze definovat pouze jednu kameru, jinak soubor nelze použít.

3.1.1 Slidy

Začátek a konec slidů určují markery na časové ose. Jejich přejmenováním na LOOP_START a LOOP_END lze docílit zacyklení času. V opač-

ném případě se na nich čas zastaví. Cykly však nelze vnořovat, překrývat, nebo navazovat přímo za sebe.

V rámci jedné komponenty je typicky více slidů. Ty jsou jednoznačně ohraničeny pomocí markerů. Markery vytváří uživatel ručně, systém na ně následně při prezentování reaguje. Marker s názvem LOOP_START označuje začátek smyčky, marker LOOP_END pak její konec. Jakýkoliv jiný název znamená, že se na něm čas zastaví. Stisknutím NEXT, nebo PREVIOUS se pak čas rozběhne, dokud opět nezastaví na nejbližším markeru.

Na markeru LOOP_START se čas zastaví pouze pokud ještě není ve smyčce. Následným stisknutím tlačítka NEXT se čas rozběhne vpřed, směrem k markeru LOOP_END. Když ho dosáhne, skočí zpět na start cyklu, ale již se nezastaví. Smyčku lze opustit opětovným stisknutím NEXT. Takoveto smyčky nelze překrývat, nebo dokonce vnořovat. Zároveň logicky nelze využít jeden marker zároveň jako konec první a začátek druhé smyčky.

3.1.2 Generování

S dopomocí plug-inu lze vytvořit šablonu, která může sloužit ke generování komponent. Šablona má formu .blend souboru, ve kterém jsou objekty označeny identifikátory. Konkrétní informace jsou do šablony dodány z textového souboru, který má formát JSON. Další možností je vygenerování komponenty z obrázků.

První možností, kterou jsem pro generování šablon implementoval, je za pomoci šablon. Uživatel si nejprve vytvoří grafickou stránku scény. Umístí do ní prázdné textové objekty, kterým nastaví konkrétní pozici, styl, materiál, nikoliv však obsah. Stejně tak lze libovolnou mesh – typicky tvar *Plane* – použít jako placeholder pro budoucí obrázek.

Pomocí záložky Template Creation v UI přiřadí uživatel objektům ve scéně identifikátory z množiny: H1, H2, OL, UL, IMAGE, NUMBER, UNASSIGNED. IMAGE lze přiřadit pouze objektu typu Mesh. Identifikátory H1, H2, OL, UL a NUMBER jsou pouze pro textové objekty. Typ UNASSIGNED není třeba přiřazovat objektům, které žádný typ z množiny nemají. Slouží pouze pro odebrání typu již přiřazenému objektu. O jednotlivých objektech a jejich přiřazených typech má uživatel přehled v UI. Když je s přiřazením spokojen, systém mu vygeneruje soubor typu JSON, který obsahuje přiřazené identifikátory objektů.

Vygenerovaný JSON následně uživatel naplní daty. U textových objektů se uvádí konkrétní textový řetězec, u obrázků cesta k souboru. Identifikátor Filename je budoucí název vygenerovaného .blend souboru. Pomocí jednoho JSON souboru lze vygenerovat více komponent, stačí vnitřní strukturu nakopírovat. Výsledný textový soubor stačí vybrat v záložce Component from Template v souboru se šablonou. Systém ji naplní daty ze zdrojového JSON

souboru a uloží jako nové Filename.blend soubory komponent. Ty lze případně dále ručně upravit, či rozšířit.

Druhou cestou, jak generovat komponenty, je pomocí připravených obrázků. Stačí je vybrat pomocí UI, zvolit jejich rozmístění a styl přechodů. Rozmístění může být v x-ové, y-ové, nebo z-ové ose, a nebo do kruhu, respektive n-úhelníku. Přechody mohou být skokové, nebo pomocí interpolace. Systém pak obrázky dle preference umístí do scény, nastaví animaci kamery a vytvoří markery. Tím je komponenta připravená k použití v prezentaci.

Takto vytvořenou komponentu lze dále upravit pouze ručně. Je proto rozumné slidy ve formě obrázků seřadit ještě před sestavením a rozmyslet si, které budou v komponent skutečně figurovat.

3.2 Prezentace

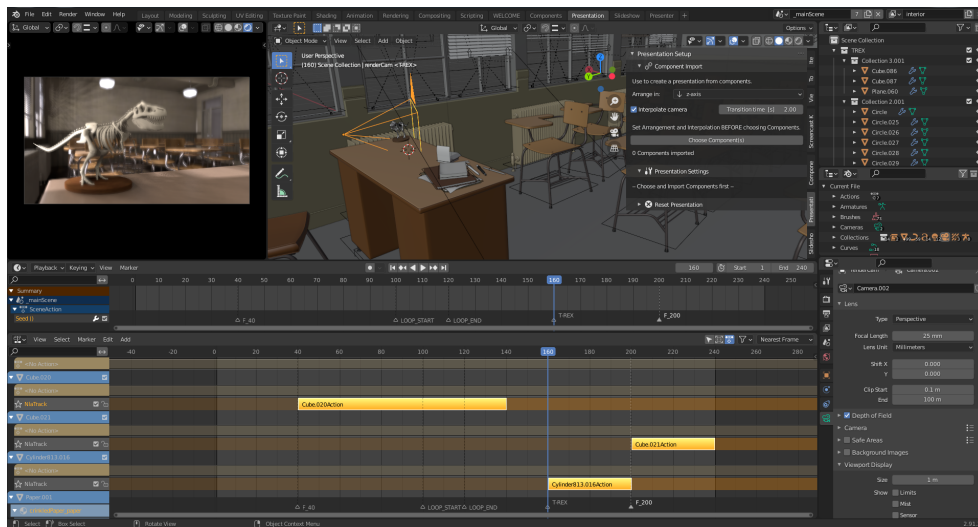
Prezentace vzniká v samostatném .blend souboru. Plug-in vytvoří kopie souborů s komponentami, přechází je a do souboru prezentace nalinkuje vybrané kolekce. Zároveň je rozmístí po scéně. Kamery komponent figurují v prezentaci jako samostatné objekty, systém je využívá k přechodům mezi komponentami. Změnit pořadí, či upravit délku přechodů lze pomocí uživatelského rozhraní NLA Editoru. Přecházení funguje díky navázání na řídicí NLA stripy pomocí driverů. Pro tyto operace je navržený workspace Presentation 3.2.

Spojením načasovaných komponent v jednom souboru vznikne prezentace připravená k prezentování. Jednu komponentu může uživatel chtít použít v různých prezentacích, proto systém před jejím přecházením vytvoří kopii. Tu systém uloží do složky umístěné a pojmenované podle .blend souboru prezentace. V případě, že by chtěl uživatel využít jednu komponentu v té samé prezentaci vícekrát, musí si sám vytvořit instance jejího souboru a ty následně použít.

Kvůli vytvoření složky pro kopie komponent, ale také kvůli otevírání dalších .blend souborů při změně časování, je třeba .blend soubor prezentace nejprve uložit. Do neuloženého souboru nelze vkládat komponenty. Pomocí skriptu lze totiž ovládat pouze jednu hlavní instanci Blenderu [32]. Plug-in tak uložený soubor prezentace zavře, poté postupně otevírá, upravuje a ukládá soubory komponent a nakonec opět otevře hlavní soubor s prezentací.

3.2.1 Vložení komponent

Vložení komponent probíhá pro uživatele ve dvou krocích – výběr a následný import. Pro systém to znamená vybrané komponenty naklonovat, vytvořit NLA stripy, přecházet, následně rozmístit komponenty ve scéně a nastavit přechody kamer.



Obrázek 3.2: Workspace Presentation

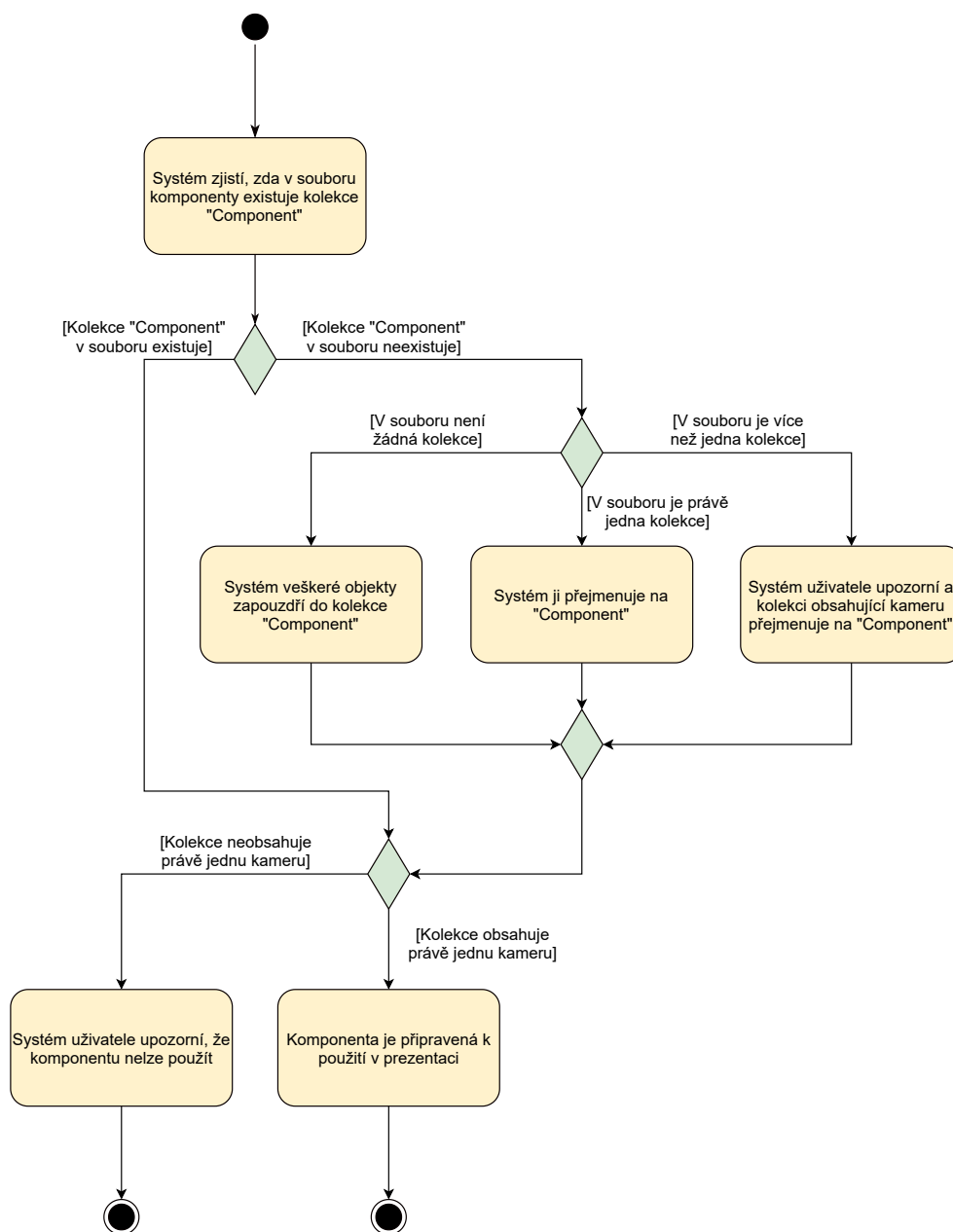
Uživatel nejprve pomocí UI plug-inu vybere rozmístění a typ přechodů mezi komponentami. Následně vybere komponenty, které chce v prezentaci použít. Tento proces může několikrát opakovat, jelikož konkrétní výběr lze provést vždy pouze v rámci jedné složky. Výběrem stvrzuje předchozí nastavení, které tak dále nelze měnit. Když je s výběrem komponent spokojený, zvolí tlačítko pro import a tím je vloží do scény.

Systém při vybírání komponent provádí následující akce:

1. Zkontroluje, zda je soubor s prezentací uložený
2. Projde zvolené komponenty a vytvoří jejich kopie
3. Jednotlivé kopie otevře a zkontroluje kolekci „Component“ (viz dále)
4. Veškeré animace přemění na NLA stripy a posune je v čase
5. Markery rovněž posune v čase a jejich název a pozici uloží do souboru markers.txt
6. Vytvoří Empty Object a k němu NLA strip, který má délku jako celá komponenta
7. Časování veškerých animací naváže pomocí driverů na Empty Object a vytvoří tak řídicí NLA strip celé komponenty

V komponentě může být více kolekcí. Systém primárně hledá kolekci z názvem „Component“, ve které by měly být objekty komponenty a právě jedna kamera. Pokud v souboru není žádná kolekce, systém ji vytvoří za uživatele.

3. IMPLEMENTACE



Obrázek 3.3: Kontrola komponenty před jejím použitím v prezentaci

Pokud je v souboru pouze jedna kolekce, zvolí systém právě tu. Pokud je v souboru více kolekcí, z nichž ani jedna nenes název „Component“, uživatele upozorní a vybere kolekci, ve které je kamera. Dále se prověřuje počet kamer v kolekci. Situaci znázorňuje obrázek 3.3. Veškeré úpravy (vytváření kolekce, přejmenování) přitom probíhají v kopii původního souboru komponenty.

Při importu kopií komponent pak systém provádí tyto akce:

1. Proveďte link přecházející kolekce „Components“
2. Proveďte append kamery z kolekce „Components“
3. Načte soubor markers.txt a vytvoří podle něj markery
4. Vytvoří animaci přechodů kamery pomocí Constraints – Copy Location, Copy Rotation a Copy Scale následující kamery
5. Z animací přechodů vytvoří NLA stripy
6. Naplní strukturu pro změnu pořadí komponent – viz Změny v prezentaci

Podrobnější informace lze nalézt v technické dokumentaci (viz. příloha), nebo v samotném zdrojovém kódu plug-inu.

3.2.2 Změny v prezentaci

Přidání a odebrání komponent do připravené prezentace je možné. Děje se tak opět pomocí UI plug-inu. Stejně tak lze změnit pořadí komponent v prezentaci – zde jsou však jistá omezení. Tyto změny se provádí v Blender záložce Nonlinear Animation (NLA Editor). Po takových změnách je však třeba přepočítat přechody kamer. Díky Overrides lze zároveň do jisté míry měnit i obsah nalinkovaných komponent.

Jak přidání, či odebrání komponenty, tak změna jejich pořadí má vliv na serializaci komponent. Je proto potřeba přepočítat přechody na nově vzniklé časové úseky mezi komponentami. K tomu systém využije připravený řídicí NLA strip každé komponenty. Aby k nim měl přístup, musí využít tzv. Overrides. Ty slouží k lokální úpravě nalinkovaných kolekcí a zároveň má díky nim systém přístup k jednotlivým objektům [33]. Jsou s nimi ale spojené komplikace popsané v podkapitole Problémy a omezení.

Pro aktivování Overrides slouží tlačítko v UI plug-inu s názvem „Override“. Je dostupné pouze pokud uživatel již vložil do scény alespoň jednu komponentu a není ve workspace Presentation. Jeho stisknutím dojde k vytvoření lokálních kopií slidů, ale také k rozřazení do nově vzniklých kolekcí pro větší přehlednost. Kolekce „COMPONENTS“ obsahuje originální kolekce nalinkovaných komponent. Kolekce „OVERRIDES“ obsahuje jejich lokální upravitelné kopie. Nakonec kolekce „TIMERS“ obsahuje kóžené řídicí Empty Objecty a jejich NLA stripy.

Po využití Overrides je tedy rovněž možná změna pořadí komponent. V NLA Editoru je vhodné použít jako filtr kolekci TIMERS, aby uživatel viděl pouze relevantní stripy. S nimi je možné hýbat myší, nebo po stisknutí klávesy G. Zároveň se v UI plug-inu zpřístupní tlačítko „Recalculate Cameras“.

To slouží právě k přepočítání přechodů a je vhodné ho použít po veškerých změnách v prezentaci.

Pokud chce uživatel přidat další komponenty do souboru prezentace, ve které již několik komponent má, vybere je opět v UI plug-inu jako v případě prázdné prezentace. Po naimportování je systém zařadí na konec prezentace. Mezi nově vloženými komponentami se kamera nastaví automaticky. Je třeba však je navázat na komponenty předchozí. K tomu opět poslouží tlačítko Recalculate Cameras.

Odstranění komponent z prezentace je možné rovněž až po použití Overrides. Slouží k tomu tlačítko v UI „Delete Component(s)“, které lze použít pouze na objekty z kolekce „COMPONENTS“. Systém odstraní komponentu, její řídicí NLA strip a markery, které k němu náležejí. Po této operaci je opět třeba přepočítat přechody kamer.

Pokud by uživatel komponentu odstranil ručně pomocí klávesy X, nebo Delete, může dojít k neočekávanému chování prezentace. V takovém případě je nejrozumnějším řešením otevřít nový soubor a vybrat komponenty znovu. Pokud by chtěl i přesto pokračovat ve stejném souboru, může použít tlačítko Reset Presentation. Po jeho stisknutí systém nastaví počet komponent na nula.

3.3 Prezentování

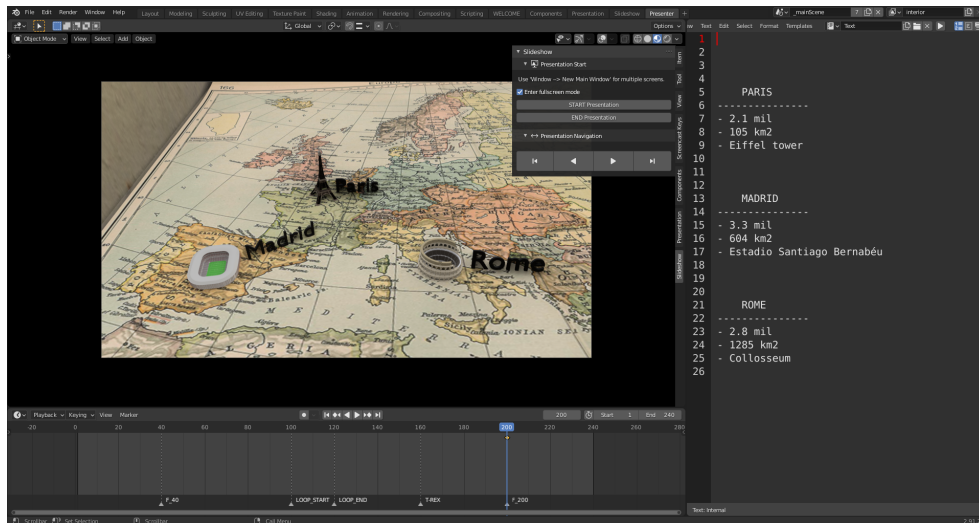
Připravená prezentace se spuštěním přepne do prezentačního workspace Slideshow. Prezentaci lze ovládat klávesnicí, nebo pomocí tlačítek v UI. Tlačítko „NEXT“ spustí čas vpřed, „PREVIOUS“ vzad. Čas se na definovaných markerech zastaví, nebo zacyklí. Změna kamery je rovněž navázána na markery. Otevřením souboru ve více oknech lze prezentovat i na více obrazovkách.

Po finalizaci prezentace přichází na řadu její odprezentování. Pro systém to v zásadě znamená každý frame zkontrolovat, kde na časové ose se nachází. Posun po ose realizuje prezentující ovládáním prezentace. Při prezentování jsou pro systém nejzásadnější markery, které určují zastavení, nebo zacyklení času – záleží na jejich názvu (viz. Slidy). Na markery na pomezí komponent jsou rovněž navázány kamery. Díky tomu se kamera automaticky přepne.

Blender umožňuje mít jeden soubor otevřený ve více oknech. Prezentující tak může mít na jedné obrazovce workspace Slideshow 3.4, který je určený pro posluchače a nabízí možnost fullscreen. Na druhé obrazovce lze přepnout na workspace Presenter 3.5, který je určený pro prezentujícího. Ten ve svém layoutu nabízí pohled na aktuální slide, dále poznámky a celou časovou osu. Prezentující má tak přehled, kde se v rámci prezentace právě nachází. Stopky pro měření délky prezentace, nebo náhled přechozího a dalšího slidu se již do implementace nedostaly.



Obrázek 3.4: Workspace Slideshow



Obrázek 3.5: Workspace Presenter

Pro prezentování se hodí render engine Eevee. Ten nenabízí tak realistické výsledky jako engine Cycles, ale pro potřeby prezentace se hodí díky své rychlosti. Prezentování tak lze provádět v okně 3D Viewport v režimu „Rendered“ a to v reálném čase [34].

3.3.1 Ovládání prezentace

Ovládání prezentace je navázáno na pohyb po časové ose. Prezentujícímu je umožněn pohyb sekvenčně vpřed a vzad a dále přeskočení na začátek, či konec prezentace. Výchozí nastavení kláves pro tyto akce je PageUp, PageDown, Home a End. Lze ho však změnit pomocí funkce Assign nebo Change Shortcut.

V uživatelském rozhraní pro ovládání prezentace jsou umístěny čtyři tlačítka v tomto pořadí: START, PREVIOUS, NEXT, END. Jejich výchozí namapování na klávesy a jejich funkce je následující:

- START (Home): Skočí na začátek prezentace (frame 1) a plynutí času zastaví.
- PREVIOUS (PageDown): Přehraje čas zpět. Pokud již čas běží, tak ho zastaví.
- NEXT (PageUp): Přehraje čas vpřed. Pokud je čas ve smyčce, skočí na její konec. Pokud již čas běží a nebyl ve smyčce, tak ho zastaví.
- END (End): Skočí na konec prezentace (end frame) a plynutí času zastaví.

Blender nabízí možnost klávesy těchto funkcí přemapovat. Stačí v UI plug-inu na vybrané tlačítko kliknout pravým tlačítkem myši a vybrat možnost *Change Shortcut*, případně *Remove* a následně *Assign Shortcut*. Poté stačí stisknout vybranou klávesu, na kterou se operace nastaví. Je však třeba si dát pozor na překrytí s výchozím nastavením Blenderu. Z tohoto důvodu nelze využít klávesy šipek pro ovládání NEXT a PREVIOUS.

Prezentaci lze ovládat i myší a tlačítka v UI plug-inu na záložce Presentation Navigation. Celé uživatelské rozhraní lze zobrazit, či skrýt v okně 3D Viewport pomocí klávesy N. Jednotlivé záložky v UI lze schovat kliknutím na šipku vedle jejich názvu.

3.4 Problémy a omezení

Hlavním problémem, na který jsem při implementaci narazil, byla změna kontextu (bpy.context). Ten je zodpovědný za chybu při použití Overrides. Chybu jsem nedokázal opravit a uživatel je tak pouze v UI informován, jak se jí vyhnout. Nevyřešeným dále zůstává kerning, který lze prozatím v Blenderu upravit pouze ručně.

Jednou ze závažnějších chyb, kterou jsem nedokázal odladit, je chyba kontextu při „nesprávném“ použití tlačítka Override. Chybu způsobuje funkce `OVERRIDE_LIBRARY_CREATE_HIERARCHY`, a to přesto, že je obalená

blokem *try* a *except*. Funkce selže z důvodů nesprávného kontextu (*bpy.context*) a způsobí pád aplikace. S tímto typem chyby jsem se při implementaci setkal několikrát, vždy však pomohlo kontext funkci předat manuálně [35]. To samé jsem vyzkoušel i zde, avšak neúspěšně.

Využití Overrides je zásadní pro úpravy prezentace a bylo tak třeba najít cestu, která by jejich použití umožnila. Ukázalo se, že chyba kontextu nastane, pokud před jejich použitím uživatel klikne do okna 3D Viewport. Tím se zřejmě kontext sám srovná a funkce neseleže. Nepochopitelnou výjimku tvoří workspace Presentation. Zde nepomůže ani tento trik. Workspace pro úpravy v prezentaci tak paradoxně neumožňuje jejich povolení. Uživatel je na tyto skutečnosti upozorněn v samotném UI, v popisu funkce Override a samozřejmě i v dokumentaci.

S úpravou prezentace souvisí ještě další omezení. Při změně pořadí je třeba posunovat v NLA Editoru vybrané řídicí stripy (TIMERS). Pokud tyto úpravy probíhají po jednom, tak je vše v pořádku. Pokud však uživatel vybere stripů více najednou, může dojít k jejich poškození. Klíčové snímky (keyframes), které jsou ukryté v jednotlivých NLA stripech, by měly mít vždy stejnou délku jako samotné stripy – koneckonců k tomu NLA stripy mimo jiné slouží [36]. Při přesouvání většího počtu však dojde k desynchronizaci. Strip má tak kratší délku než reálná animace. K opětovné synchronizaci dojde až při vybrání stripu a přepnutí do Edit modu (Tab).

Posledním nevyřešeným problémem je kerning. Ten je totiž v Blenderu interně definován jako kladné číslo. To způsobuje, že pomocí skriptu, lze mezeru mezi písmeny pouze zvětšit. Pokoušel jsem se navázat na skripty *kern-dump* [37], které s kerningem pracují, ale neúspěšně. Dokonce jsem narazil na řešení z roku 2016 určené přímo pro zdrojový kód Blenderu [38]. Přes drobné komplikace proběhla následná kompilace Blenderu správně. Kerning však zůstal nezměněný. Na mé dotazy na Blender fóru bohužel také nikdo nezareagoval. Kerning tak zůstává předmětem budoucích rozšíření.

3.5 Možnosti rozšíření

Vzhledem k jednotnému typu souboru komponent, by mohly být v budoucnu přidány další možnosti generování těchto dílčích částí. Kerning by mohl být vyřešen externím skriptem. Dále lze do systému přidat další rozmístění komponent a slidů. Také přechody mezi komponentami by mohly být nastavitelné – namísto jednoduché interpolace. Workspaces by mohly být vytvářeny samotným plug-inem místo využití .blend šablony. Nové možnosti by pak přinesla interaktivita prezentace.

V práci jsem prověřil možnosti generování 3D komponent a prezentací. Jejich implementace ve větší míře však vybočovala z rozsahu, který byl práci stanoven. Další cesty, jak komponenty, či prezentace generovat, by tak mohly

být součástí budoucích snah. Podmínkou pro jejich úspěšné použití je formát .blend a markery k definici slidů.

Blender zatím neumí pracovat s kerningem textu u textových objektů ve scéně. V uživatelském rozhraní to ale paradoxně zvládá. Sami vývojáři Blenderu uvádí, že „se nejedná o bug, ale téma kerningu zůstává otevřené“ [39]. Do té doby by mohl pomoci externí skript, který by kerning spočítal a aplikoval.

V současné době nabízí plug-in čtyři možnosti rozmístění komponent: ve směru kladné x-ové osy, kladné y-ové osy, záporné z-ové osy, nebo do n-úhelníku. Ty by mohly být dále rozšířeny. Stejně tak by mohla být přidána možnost nastavení přechodu kamery mezi komponentami. Místo jednoduché interpolace by se kamera mohla pohybovat po upravitelné křivce. Obnášelo by to však její úpravy při každé změně pozice komponenty.

Aby měl uživatel přístup k připraveným workspaces, je potřeba v souboru PRESENTATION_TEMPLATE.blend zavolat funkci *Save Startup File*. Plug-in by tak bylo vhodné doplnit o funkci, která tyto workspaces vytvoří při instalaci. Tím by odpadla nutnost stažení této šablony. Workspace Presenter by pak mohl být rozšířen o stopky, nebo náhled předchozího a následujícího slidu. Zobrazené poznámky by mohl plug-in načítat jako samostatný textový soubor pro každou komponentu.

Nakonec by mohla být prezentace rozšířena o prvek interaktivity. Ten by přidal možnost nelieárního pohybu - podobně jako u prezentačního systému Prezi. Kliknutím na objekt by prezentace přeskočila na vybranou komponentu, mohla by spustit jeho animace apod.

Testování

V rámci testování jsem se zabýval pokrytím navržených případů užití. Míra pokrytí se ukázala jako dostatečná. Dále jsem provedl uživatelské testování. Při něm se ukázaly známé chyby při úpravě prezentace. Ty však neznemožnily dokončení testovacího scénáře. Pouze u naplnění šablony pomocí souboru JSON došlo k větším komplikacím. Většina dalších drobných chyb, které se při testování vyskytly, byla následně opravena.

4.1 Pokrytí případů užití

Případy užití jsem navrhl v rámci kapitoly Analýza uživatelů. V této kapitole jsem zhodnotil míru jejich pokrytí. Případ užití pro synchronní a asynchronní procházení prezentace byl vyřazen již v návrhu, a proto zde nefiguruje. Ostatní případy užití dopadly ve velké míře uspokojivě. Výstup implementace tak lze prohlásit za prezentační systém.

Pro připomenutí uvádím kompletní seznam případů užití, bez vyřazeného UC9 – ten byl shledán mimo rozsah této práce. Poté následuje zhodnocení míry naplnění každého případu užití.

- UC1: Vytvoření šablony slidu
- UC2: Naplnění šablony daty
- UC3: Spojení slidů do prezentace
- UC4: Úprava slidu v prezentaci
- UC5: Přidání/odebrání slidu
- UC6: Změna pořadí slidů
- UC7: Spuštění připravené prezentace

- UC8: Přepínání slidů

UC1: Vytvoření šablony slidu

Šablony slidů, respektive komponent, je možné vytvářet s dopomocí plug-inu. Uživatel vytvoří libovolnou scénu a jejím objektům přiřadí identifikátory jako H1, IMAGE apod. Objekty slouží jako placeholdery. Díky tomu lze oddělit grafický návrh komponent od následného naplnění daty, což byl hlavní účel tohoto případu užití. Generování šablon, které popisuje alternativní scénář tohoto případu užití, se do implementace nedostalo.

UC2: Naplnění šablony daty

Připravenou šablonu lze naplnit daty. To může probíhat ručně, tedy prostým přepsáním textu v souboru šablony a doplněním konkrétních obrázků. Alternativně lze použít soubor JSON a vygenerovat komponentu pomocí něj. Strukturu tohoto textového souboru vytvoří plug-in za uživatele. Pomocí jednoho JSON souboru je možné dokonce vygenerovat více komponent. V tomto případě je tak splněn hlavní i alternativní scénář užití.

UC3: Spojení slidů do prezentace

Jednotlivé komponenty, které v sobě obsahují definované slidy, lze spojit do výsledné prezentace. Jak komponenty, tak prezentace mají formu .blend souborů. Komponenty uživatel vybere přes UI plug-inu, ten je správně přechasuje, rozmístí ve scéně a nastaví mezi nimi přechody. Jediným omezením je fakt, že nastavení rozmístění a přechodů již po vybrání první komponenty nelze změnit. Komponenty se v kontextu prezentace chovají jako celky, lze s nimi libovolně hýbat, rotovat, zvětšovat je a zmenšovat.

UC4: Úprava slidu v prezentaci

Komponenty a jejich slidy jsou v souboru prezentace nalinkované. Systém zároveň vytváří kopii původního souboru, ve které provádí změny před vložením do prezentace. Stačí tak změnit tuto kopii originální komponenty, která se nachází ve složce pojmenované podle souboru prezentace. Díky vazbě se změny projeví i v souboru s prezentací. Zároveň vznikla možnost úprav komponent přímo v prezentaci po použití Overrides. Tyto změny jsou pouze lokální a neovlivní tak původní soubor, ani jeho kopii.

UC5: Přidání/odebrání slidu

Přidání další komponenty – a tím i slidů – je možné přes stejné UI jako jejich prvotní vložení. Nově přidané komponenty jsou automaticky zařazeny na konec prezentace. Je však třeba je navázat na předchozí část prezentace pomocí Recalculate Cameras. U rozmístění na osách jsou nově přidané komponenty umístěny správně ve směru dané osy. U tvaru n -úhelníku k vytvoření $(n+1)$ -úhelníku nedojde, je potřeba to udělat ručně. Pro odebrání komponent z prezentace existuje samostatné tlačítko v UI Delete Component(s). Odstranění

je možné až po použití Overrides. Po takovéto změně je opět třeba přepočítat kamery. Problémem je odstranění komponenty pomocí klávesy Delete, nebo X. Takováto změna se nepropíše do datové struktury systému a může způsobit chyby.

UC6: Změna pořadí slidů

Slidy jsou definovány uvnitř komponenty a změna jejich pořadí se děje tak právě uvnitř souboru komponenty. Změna pořadí celých komponent v prezentaci je možná, ale může být problematická. Využívají se zde Overrides, které mohou způsobit pád aplikace (viz. Problémy a omezení). Nicméně při dodržení určitých podmínek je změna možná. Overrides do prezentace vloží řídicí NLA stripy jednotlivých komponent (TIMERS), se kterými lze v rámci NLA Editoru hýbat – nejlépe po jednom. Změnit pořadí pak znamená seřazení NLA stripů za sebe. Statické komponenty jsou reprezentovány jako NLA stripy o 1 framu. Pro všechny tyto změny je vhodný workspace Presentation v příloženém souboru PRESENTATION_TEMPLATE.blend.

UC7: Spuštění připravené prezentace

Jeden .blend soubor lze otevřít ve více oknech a prezentaci tak může uživatel spustit na druhé obrazovce. Spuštění se nachází v UI Slideshow. Systém se pokusí přepnout do stejnojmenného workspace, který je ve zmiňované šabloně PRESENTATION_TEMPLATE.blend. Pokud workspace nenalezne, přepne do workspace Layout. Klávesová zkratka pro spuštění prezentace je F5. Pokud by chtěl uživatel připravenou prezentaci přenést do jiného zařízení a spustit ji tam, je potřeba zkopírovat si i složku nesoucí jméno .blend souboru s prezentací. Ta obsahuje nalinkované komponenty. Samozřejmostí je pak samotný plug-in.

UC8: Přepínání slidů

Spuštěná prezentace je ovladatelná pomocí tlačítek v UI a pomocí klávesnice. Pomocí NEXT (PageUp) a PREVIOUS (PageDown) je realizován pohyb po časové ose. Tím dochází k přepínání slidů. Zároveň je možné kdykoliv přeskočit na začátek, či konec prezentace pomocí Home a End. Veškeré toto nastavení kláves si může uživatel změnit. Přeskočení na vybraný slide se do implementace nedostalo.

S výjimkou alternativního scénáře UC1, tedy generování šablon, pokrývá implementovaný systém z velké části veškeré případy užití. V dostatečné míře tak naplňuje požadavky uživatelů a lze ho prohlásit za prezentační systém využívající 3D.

4.2 Uživatelské testování

Pro uživatelské testování jsem navrhl celkem devět testovacích scénářů, které prověřily jak funkčnost systému, tak jeho uživatelské rozhraní. Toto testování použitelnosti probíhalo distančně, zúčastnili se ho postupně tři vybraní dobrovolníci. U některých scénářů měli testeři menší problémy s UI, u jiných se ukázaly implementační nedostatky. Celkově však testování dopadlo uspokojivě a většina těchto chyb byla následně opravena.

Vzhledem k současné situaci spojené s pandemií koronaviru jsem uživatelské testování provedl distanční formou. Konkrétně se jednalo o videohovor se sdílením obrazovky. Měl jsem tak možnost sledovat jak verbální a neverbální komunikaci testerů, tak jejich obsluhování systému. Zároveň mi add-on *Screen-cast Keys* [40] pomohl k tomu, abych v danou chvíli mohl pozorovat i konkrétní stisknutí kláves.

Každý tester prošel celkem devět testovacích scénářů cílených na UI i funkčnost plug-inu. K dispozici měl samotný plug-in, šablonu pro práci s ním (`PRESENTATION_TEMPLATE.blend`), textový soubor `welcome.txt`, který se zobrazí po otevření šablony, add-on `screen-cast-keys.zip` a složku s ukázkami `.blend` komponent a jejich šablon.

Tři vybraní testeři splňovali profil uživatele, který umí pracovat s aplikací Blender v takové míře, aby byl schopen vytvořit vlastní animovanou 3D scénu. Zároveň se jednalo o uživatele, kteří v nedávné době použili prezentační systém pro účely vytvoření prezentace a jejího odprezentování. Všichni tři testeři používali operační systém Windows 10 a Blender verze 2.90, nebo novější.

Pro hodnocení uživatelského rozhraní jsem použil tuto stupnici:

- 100 %** Testující byl schopen projít celým scénářem bez zásahu moderátora a neměl k UI výhrady.
- 75 %** Testující prošel scénářem s drobnou výpomocí moderátora a nebo měl k UI drobné výhrady.
- 50 %** Testující prošel scénářem s častými zásahy moderátora a nebo měl výhrady k UI.
- 25 %** Testující dokončil scénář pouze zásluhou moderátora, který v celém průběhu scénáře poskytoval instrukce „krok po kroku“ a nebo měl tester zásadní výhrady k UI.
- 0 %** Testující nebyl schopen dokončit scénář a nebo správné UI vůbec nenašel.

Pro hodnocení funkčnosti jsem použil tuto stupnici:

- 100 %** Veškeré testované funkcionality fungovaly bez problémů a korektně.

- 75 %** U některých funkcionalit se objevily drobné komplikace, které ovšem nenarušily průběh scénáře a bylo dosaženo očekávaných výsledků.
- 50 %** U některých funkcionalit se objevily závažnější komplikace, ale podařilo se dosáhnout konce scénáře.
- 25 %** U většiny funkcionalit se objevily závažnější komplikace, ale podařilo se dosáhnout konce scénáře.
- 0 %** V průběhu testování se objevily chyby, které znemožnily dokončení scénáře.

Následuje výčet jednotlivých testovacích scénářů. U každého je uveden název, jeho účel a typ – zda je zaměřen na uživatelské rozhraní (UI) nebo funkcionality (F). Dále jsou uvedeny podmínky pro jeho provedení, kroky k jeho provedení a očekávaný výsledek. Nakonec je každý testovací scénář doplněn slovním hodnocením a posouzením výsledku. Procentuální ohodnocení je vypočítáno jako průměr hodnocení všech tří testů (1 tester = 1 test). K testerům je dále referováno jako tester A, tester B a tester C.

T1

Název:	Seznámení se šablonou prezentace
Účel:	Prvotní orientace uživatele v souboru šablony prezentace
Typ testu:	UI
Podmínky:	- Nezměněná šablona PRESENTATION_TEMPLATE.blend
Kroky:	1. Uživatel otevře soubor PRESENTATION_TEMPLATE.blend 2. Přečte si úvodní informace 3. Zorientujte se v nových workspaces
Očekávaný výsledek:	Uživatel ví, k čemu slouží jednotlivé workspaces a je připraven využít plug-in.
Hodnocení výsledku:	UI: $(75 + 100 + 75)/3 = 83,3 \%$

Pro přečtení úvodních informací v šabloně, byli testeři schopni ve velké míře svými slovy popsat strukturu 3D prezentace (slide, komponenta atd.). Tester A neznal pojem workspace, po objasnění však již neměl problémy, stejně jako zbylí. Tester C pak nejprve myslel, že workspace Presentation slouží k prezentování, místo vytváření prezentace. Dále se testeři shodli, že instruktážní

4. TESTOVÁNÍ

text v šabloně je vyčerpávající a četli tak pouze úvodní shrnující odstavce. Jinak se prvotní orientace v souboru šablony zdála jako přímočará.

T2

Název:	Vytváření komponent – obrázky
Účel:	Test UI pro generování komponent a test funkcionality generování komponent z obrázků
Typ testu:	UI, F
Podmínky:	- Soubor PRESENTATION_TEMPLATE.blend - Obrázky pro vytvoření komponenty
Kroky:	1. Uživatel přepne do workspace Component 2. Pomocí UI vybere alespoň 5 obrázků 3. Sestaví z nich komponentu obrázků pod sebou s vypnutou interpolací kamery 4. Komponentu uloží pro použití v prezentaci
Očekávaný výsledek:	Uživatel vytvořil zadanou komponentu a uložil ji jako samostatný .blend soubor.
Hodnocení výsledku:	- UI: $(100+75+75)/3 = 83,3 \%$ - F: $(75+75+100)/3 = 83,3 \%$

U druhého testovacího scénáře bylo úkolem vytvořit komponentu z obrázků. Co se týče UI, tak tester B nejprve využíval záložku pro vytváření šablon, poté se však sám opravil. Tester C pak myslel, že sestavení obrázků pod sebe znamená v y-ové ose.

U funkcionalit se ukázalo, že je třeba mít povolený plug-in *Import Images as Planes*, aby import fungoval. Druhý tester pak nejprve importoval obrázky po jednom. Po nápodě, že je třeba vybrat všechny a až poté importovat, ale scénář splnil.

T3

Název:	Vytváření komponent – vytvoření šablony
Účel:	Test UI pro generování komponent a test funkcionality vytváření šablon pro komponenty
Typ testu:	UI, F
Podmínky:	- Soubor PRESENTATION_TEMPLATE.blend

Kroky:	<ol style="list-style-type: none"> 1. Uživatel přepne do workspace Component 2. Vytvoří scénu alespoň s 3 textovými objekty, jedním obrázkem a alespoň 2 objekty pozadí 3. Objektům přiřadí pomocí UI správný typ 4. Vygeneruje soubor JSON 5. Soubor s šablonou uloží
Očekávaný výsledek:	Uživatel vytvořil šablonu komponenty a uložil ji jako samostatný .blend soubor. Dále vygeneroval JSON soubor pro její naplnění.
Hodnocení výsledku:	<ul style="list-style-type: none"> - UI: $(50+75+100)/3 = 75,0 \%$ - F: $(75+75+100)/3 = 83,3 \%$

S uživatelským rozhraním pro vytváření šablon měl menší problém pouze tester A. Při vytváření scény použil pro obrázek UI Image Slides, které slouží k vytváření samostatných komponent z obrázků. V tomto ohledu ho tak bylo třeba navést.

Všichni po vytvoření jednoduché scény správně přiřadili objektům jejich typy. Jedinou chybou bylo ve dvou případech opomenutí kamery, která musí být součástí každé komponenty. Jeden z testerů pak hledal v seznamu typ objektu pro pozadí. Když si však znovu přečetl přiložený textový soubor, pochopil, že pozadím jsou neoznačené objekty. Vygenerování souboru JSON pak proběhlo bez problémů.

T4

Název:	Vytváření komponent – naplnění šablony
Účel:	Test funkcionality generování komponent ze šablony a jejího UI
Typ testu:	UI, F
Podmínky:	<ul style="list-style-type: none"> - Soubor PRESENTATION_TEMPLATE.blend - Šablona komponenty - Vygenerovaný JSON soubor
Kroky:	<ol style="list-style-type: none"> 1. Uživatel naplní JSON soubor daty alespoň pro 2 komponenty 2. Otevře soubor se šablonou 3. Pomocí UI vybere JSON soubor s daty 4. Vygeneruje alespoň 2 komponenty 5. Vytvořené komponenty otevře a zkontroluje
Očekávaný výsledek:	Uživatel ze šablony vygeneroval alespoň 2 komponenty pro prezentaci.

4. TESTOVÁNÍ

Hodnocení výsledku: - UI: $(50+50+75) = 58,3 \%$
- F: $(0+0+50) = 16,7 \%$

Naplnění šablony z předchozího scénáře nedopadlo dobře. U uživatelského rozhraní měli testeři výhrady k zadávání cesty k obrázku. Nejčastěji zde znělo slovo „otravné“. Tester A pak zadal cestu pouze s jedním oddělovačem „\“ a import obrázku by tak následně nefungoval. Tester B měl problém obecně s formátem JSON, se kterým neměl zkušenosti.

Scénář ale bohužel k vygenerování komponent v prvních dvou případech vůbec nedošel. Plug-inu se totiž nepodařilo načíst vybraný JSON soubor. Až u testera C se po několika pokusech a ručním přepsání cesty k souboru povedlo komponenty vygenerovat.

T5

Název: Vytváření komponent – ruční vytvoření komponenty

Účel: Test pochopení struktury komponenty a jejího ručního vytváření

Typ testu: UI, F

Podmínky: - Soubor PRESENTATION_TEMPLATE.blend

Kroky: 1. Uživatel si přečte návod pro vytváření komponent
2. Inspiruje se příloženou ukázkou komponenty
3. Vytvoří komponentu o alespoň 3 slidech, 1 smyčce, s animovanou kamerou, 1 animovaným objektem a 2 statickými objekty
4. Komponentu uloží jako samostatný .blend soubor

Očekávaný výsledek: Uživatel ručně vytvořil animovanou komponentu připravenou pro prezentaci.

Hodnocení výsledku: - UI: $(75+75+75)/3 = 75,0 \%$
- F: $(100+100+75)/3 = 91,7 \%$

Při tomto scénáři testeři potvrdili, že jim jako lepší předloha pro ruční vytvoření komponenty přišel ukázkový soubor Component_3D_Example.blend, spíše než psaný návod. U některých bylo třeba pouze připomenout kameru, nebo použití markerů. Všichni tak vytvořili korektní animovanou komponentu. Většinový názor byl, že vytváření je zdlouhavější než ve 2D, ale výsledek vypadá atraktivněji.

Také padla myšlenka, že přejmenování markerů je v UI příliš ukryté. U testera C pak při použití PLAY v rozhraní timeline nefungovalo zastavení na markerech. Po použití PageUp však již vše fungovalo, jak mělo.

T6

Název:	Prezentace – spojení komponent
Účel:	Test UI pro vytváření prezentace a funkcionality přecházení a importování komponent
Typ testu:	UI, F
Podmínky:	- Soubor PRESENTATION_TEMPLATE.blend - Alespoň 4 připravené komponenty
Kroky:	<ol style="list-style-type: none"> 1. Uživatel přepne do workspace Presentation 2. Nastaví rozmístění komponent za sebe a interpolaci kamery na jednovteřinovou 3. Vybere alespoň 4 komponenty prezentace 4. Sestaví z nich výslednou prezentaci 5. Soubor s prezentací uloží
Očekávaný výsledek:	Uživatel vytvořil prezentaci o 4+ komponentách, které jsou umístěny v y-ové ose a je mezi nimi 1s interpolace.
Hodnocení výsledku:	- UI: $(75+75+50)/3 = 66,7 \%$ - F: $(100+75+50)/3 = 75,0 \%$

Pro spojení komponent ani jeden z testerů nevyužil workspace Presentation, což v zásadě ale není chyba. Všichni správně zvolili UI záložku Component Import. Někteří opět váhali, kterou osu zvolit, aby komponenty byly za sebou.

Tester C měl výhrady, že nastavení prezentace nejde změnit již po vybrání první komponenty. Zároveň tím, že komponenty vybíral postupně, tak následně nefungoval správně import markerů z textového souboru – ten se totiž pokaždé smazal. Bez markerů nefungovaly ani přechody kamery. Bylo tak třeba komponenty vybrat najednou, aby sestavení fungovalo správně.

T7

Název:	Prezentace – úprava
Účel:	Test UI pro úpravu prezentace a funkcionalit změna pořadí a odebrání komponent
Typ testu:	UI, F

4. TESTOVÁNÍ

- Podmínky:** - Soubor PRESENTATION_TEMPLATE.blend
- Prezentace s 4+ komponentami
- Kroky:** 1. Uživatel přepne do workspace Presentation
2. Povolí změny v prezentaci
3. Udělá 2 změny v pořadí
4. Uparví obsah 1 komponenty (lokálně)
5. Odebere 1 komponentu
- Očekávaný výsledek:** Uživatel změnil pořadí komponent, jedné změnil obsah a jednu z prezentace odebral.
- Hodnocení výsledku:** - UI: $(50+25+50)/3 = 41,7 \%$
- F: $(50+50+75)/3 = 58,3 \%$

Změny v prezentaci vyžadují použití Overrides. Ty již v implementaci ukázaly, že mohou způsobit chybu Blenderu a celý program shodit (viz. Problémy a omezení v kapitole Implementace). Přesně to se také ve dvou ze tří případů stalo. Testeři byli upozorněni, že jde o známou chybu, která je popsána v příloženém textovém souboru. Poté již chyba nenastala. Vyskytl se pouze drobný problém s kolekcí “COMPONENTS” – ta se správně vytvořila, ale již do sebe nezahrnula nalinkované komponenty.

K uživatelskému rozhraní měli testeři několik připomínek. Především byli zmateni, co se vlastně po použití Overrides stalo a mnohdy se až polekali, že prezentaci rozbili. Bylo tak třeba je nasměrovat do workspace Presentation, které by opět žádný z nich nepoužil, a vybídnout k opětovnému přečtení dokumentace. I přesto si tester B nevěděl rady se změnou pořadí. Lokální úpravy komponenty rovněž neproběhly hladce a bylo třeba ho navést.

T8

- Název:** Prezentování – spuštění na více obrazovkách
- Účel:** Test UI pro spuštění prezentace na více obrazovkách a funkcionalit s tím spojených
- Typ testu:** UI, F
- Podmínky:** - Soubor PRESENTATION_TEMPLATE.blend
- Prezentace s 3+ komponentami
- Kroky:** 1. Uživatel otevře nové okno v souboru s prezentací
2. V jednom okně otevře workspace Presenter
3. V druhém okně spustí prezentaci na celou obrazovku (workspace Slideshow)

Očekávaný výsledek: Uživatel spustil prezentaci na dvou obrazovkách, na každé má jiný workspace.

Hodnocení výsledku: - UI: $(100+75+100)/3 = 91,7 \%$
- F: $(100+100+100)/3 = 100 \%$

Ani jeden z testerů neměl více obrazovek, scénář byl tak simulován pouze pomocí více oken aplikace Blender. UI se jevilo jako bezproblémové, všichni správně otevřeli další okno s prezentací. Pouze tester B použil místo New Main Window pouze funkci New Window a zprvu tak neměl přístup k jiným workspaces.

Jinak spuštění prezentace proběhlo bez problémů. Systém správně nastavil Viewport Shading a fullscreen. Kvůli absenci druhé obrazovky samozřejmě nebylo možné, mít prezentaci na celou obrazovku a zároveň vidět workspace Slideshow i Presenter.

T9

Název: Prezentování – ovládání a ukončení prezentace

Účel: Test UI pro ovládání, funkcionality ovládání klávesnicí a změny namapování tlačítek

Typ testu: UI, F

Podmínky: - Soubor PRESENTATION_TEMPLATE.blend
- Spuštěná prezentace s 3+ komponentami

Kroky: 1. Uživatel ovládá prezentaci pomocí UI plug-inu
2. Uživatel ovládá prezentaci pomocí klávesnice
3. Změní namapování tlačítka NEXT na libovolnou nevyužitou klávesu
4. Nakonec prezentaci ukončí

Očekávaný výsledek: Uživatel ovládal prezentaci jak pomocí myši a UI, tak pomocí klávesnice. Ovládání klávesnicí změnil a prezentaci úspěšně ukončil.

Hodnocení výsledku: - UI: $(100+75+75)/3 = 83,3 \%$
- F: $(75+75+75)/3 = 75 \%$

Samotné prezentování proběhlo hladce. Pro přepínání slidů využívali testéři nejprve myš a UI plug-inu. Po zjištění, že lze použít i klávesnici, ale většinou zůstali právě u varianty PageUp, PageDown. Tester B se prezentaci pokoušel ovládat pomocí šipek, které slouží pouze k posunu o jeden frame.

4. TESTOVÁNÍ

Tester C hledal změnu namapování tlačítek nejprve v prostředí 3D Viewport místo UI plug-inu.

Všichni tři testeři se pokusili změnit přepínání slidů na levou a pravou šipku. Po objasnění, že tyto klávesy jsou v Blenderu využity pro jinou akci a nelze je tak použít, zvolili jiné klávesy. Ukončení prezentace proběhlo téměř bez problémů. Tester A se o ukončení pokoušel klávesou Escape, kterou však nelze použít ze stejného důvodu, jako šipky.

Celkové hodnocení všech testovacích scénářů zachycuje tabulka 4.1:

Tabulka 4.1: Výsledky uživatelského testování

	T1	T2	T3	T4	T5	T6	T7	T8	T9
UI	83,3 %	83,3 %	75 %	58,3 %	75 %	66,7 %	41,7 %	91,7 %	83,3 %
F	X	83,3 %	83,3 %	16,7 %	91,7 %	75 %	58,3 %	100 %	75 %

Následuje shrnutí problémy, které z testování vzešly a jejich případné řešení:

- Špatná orientace ve workspaces, nebo při využívání plug-inu
 - Vyřešeno. Doplnění instruktážního videa na stránky s dokumentací a přehlednější úvodní text v šabloně PRESENTATION_TEMPLATE.blend.
- Zaměnění y-ové a z-ové osy při rozmístění obrázků a komponent
 - Vyřešeno. UI pro rozmístění slidů bylo doplněno o ikony znázorňující směr.
- Nenainstalovaný add-on *Import Images As Planes*
 - Vyřešeno. Plug-in ho nainstaluje za uživatele.
- Nemožnost načtení JSON souboru
 - Chybu se bohužel nepodařilo znovu nasimulovat a tudíž ani opravit. Byla pouze přidána varovná hláška, která by mohla pomoci problém vyřešit ručně.
- Nefunkční import markerů při postupném vybírání komponent
 - Vyřešeno. Soubor s markery je smazán až po samotném naimportování komponent.
- Nevyužívání Presentation workspace ani po použití Overrides

- Vyřešeno. Plug-in sám přepne do workspace Presentation po použití Overrides.
- Kolekce COMPONENT neobsahovala komponenty
 - Vyřešeno. Problém nastával, pokud při importu komponent v souboru již existovala nějaká kolekce.
- Namapování na již využitá tlačítka
 - Tento problém nebyl dále řešen.

Závěr

Cílem této bakalářské práce bylo vytvořit prezentační systém založený na aplikaci Blender, který umožní prezentace vytvářet i prezentovat. Po analýze vznikl návrh systému, který z těchto požadavků vycházel. Následně jsem implementoval výsledný plug-in. Testování ukázalo, že splňuje potřeby prezentačního systému a cíl práce byl tak splněn. Nakonec jsem navrhl možnosti dalšího rozšíření.

V práci jsem se zabýval analýzou existujících prezentačních systémů a existujících rozšíření Blenderu, které se věnují prezentaci. Dále jsem zanalyzoval strukturu, obsah a vizuální složku prezentací. Z analýzy uživatelů vzešly požadavky na výsledný prezentační systém. Vytvořil jsem tak několik případů užití, které tyto požadavky pokryly. Během analýzy jsem identifikoval tři roviny - rovina vytváření slidů, či jiných dílčích částí prezentace, rovina spojování těchto částí do výsledné prezentace a rovina samotného prezentování.

Následně jsem vytvořil návrh prezentačního systému, který zohledňoval tyto tři roviny a umožňoval je realizovat ve 3D. Návrh počítal se strukturou prezentace do trojrozměrných komponent, v rámci kterých lze definovat jednotlivé slidy. Spojením těchto komponent v jedné 3D scéně vznikla 3D prezentace. V návrhu nechybělo ani spuštění prezentace, její ovládání a ukončení. Všechny tyto procesy byly přitom úzce provázány s časovou osou - skládání komponent znamenalo přecházet animace, prezentovat znamenalo pohybovat se po časové ose dopředu, či dozadu. Dále jsem navrhl uživatelská rozhraní a rozložení obrazovky (layout) pro jednotlivé procesy spojené s prezentací.

Při implementaci pak tyto představy nabraly konkrétních tvarů a forem. Pro reprezentaci komponent byl zvolen formát .blend, který dovoluje uchovávat jak 3D obsah, tak metadata o kameře, či jednotlivých slidech. Kromě ručního vytváření komponent jsem uživateli dal i jednoduchou možnost jejich generování. Rozmístěním komponent v jedné scéně a nastavením přechodů vznikla prezentace. Po tomto spojení komponent byly umožněny další úpravy

prezentace - přidání dalších komponent, odstranění existujících, či změna jejich pořadí. Prezentujícímu bylo poskytnuto ovládání prezentace pomocí myši a UI plug-inu i pomocí klávesnice. Prezentování navíc může probíhat na více obrazovkách.

Po dokončení implementace následovalo testování. Zhodnotil jsem v něm míru pokrytí navržených případů užití - ta se ukázala jako uspokojivá. Jedinou pomyslnou černou tečku tvořila omezení při úpravě prezentace. Nakonec proběhlo i uživatelské testování, které prověřilo jak funkčnost systému, tak jeho uživatelské rozhraní. Většina nedostatků, které z testování vzešly, byla následně opravena.

Ukázalo se, že výsledný plug-in ve spojení s Blenderem v dostatečné míře splňuje veškeré požadavky uživatelů a lze ho tak prohlásit za prezentační systém. Tím bylo dosaženo cíle této práce. Výstup představuje alternativu pro prezentující, kteří hledají nový, atraktivnější způsob prezentování, který jejich posluchače zaujme. Díky navázání na časovou osu a využití časových smyček je možné vytvářet daleko dynamičtější prezentace. Dalším specifickým je rozdělení prezentace do dílčích komponent, které jsou vytvářeny samostatně a následně serializovány.

V budoucnu by mohly být doplněny další možnosti generování komponent, nebo větší personalizace přechodů kamery. Dále by mohla být přidána podpora kerningu, nebo odstraněny omezení spojené s úpravou prezentace. Workspace Presenter by bylo možné doplnit o další funkce z návrhu jako stopky, či náhledy sousedních komponent. Nakonec by do celé prezentace mohla přibýt interaktivita objektů.

Literatura

- [1] VISUAL SOFTWARE SYSTEMS Ltd. *Emaze* [online]. 2020, [cit 05.12.2020]. Dostupné z: <https://www.emaze.com/presentations/>
- [2] BLENDER FOUNDATION. *Blender 2.91* [software]. 25.11.2020, [cit 03.12.2020]. Dostupné z: <https://www.blender.org/>, (překlad vlastní)
- [3] MICROSOFT CORP. *PowerPoint* [software]. 2019, [cit 04.12.2020]. Dostupné z: <https://www.microsoft.com/cs-cz/microsoft-365/powerpoint>
- [4] POWTOON, Ltd. *Powtoon* [online]. 2012, [cit 12.12.2020]. Dostupné z: <https://www.powtoon.com/>
- [5] PAVLOVA, Iveta. *7 Most Popular Software for Presentations* [online]. 2019 [cit. 15.02.2021]. Dostupné z: <https://graphicmama.com/blog/7-popular-software-presentations/>, (překlad vlastní)
- [6] THE APACHE SOFTWARE FOUNDATION. *OpenOffice Impress* [software]. 2018, [cit 04.12.2020]. Dostupné z: <https://www.openoffice.cz/impress>
- [7] GOOGLE LLC. *Google Slides* [online]. 2006, [cit 08.12.2020]. Dostupné z: <https://www.google.com/slides/about/>
- [8] TANTAU Till, WRIGHT Joseph, MILETIĆ Vedran. *Beamer - A LaTeX class for producing presentations* [online]. 2003, [cit 21.01.2021]. Dostupné z: <https://ctan.org/pkg/beamer>
- [9] CANVA, Pty Ltd. *Canva* [online]. 2012, [cit 21.01.2021]. Dostupné z: <https://www.canva.com/>
- [10] SLIDEBEAN, Inc. *Slidebean* [online]. 2014, [cit 22.01.2021]. Dostupné z: <https://slidebean.com/>

- [11] PERRON, Brain E. a Alyson G. STEARNS. *A Review of a Presentation Technology: Prezi* [online]. 2010 [cit. 10.12.2020]. Dostupné z: doi:10.1177/1049731510390700, (překlad vlastní)
- [12] PREZI, Inc. *Prezi* [online]. 2009, [cit 10.12.2020]. Dostupné z: <https://prezi.com/>
- [13] EL HATTAB, Hakim. *Reveal.js* [online]. 2017, [cit 10.12.2020]. Dostupné z: <https://revealjs.com/>
- [14] SAVATON, Guillaume. *Sozi* [online]. 2010, [cit 10.12.2020]. Dostupné z: <https://sozi.baierouge.fr/>
- [15] INKSCAPE. *Inkscape: Open Source Scalable Vector Graphics Editor* [software]. 2003, [cit 10.12.2020]. Dostupné z: <https://inkscape.org/>
- [16] ANIMAKER, Inc. *Animaker* [online]. 2014, [cit 12.12.2020]. Dostupné z: <https://www.animaker.com/>
- [17] BRYXEN, Inc. *Animaker* [online]. 2016, [cit 12.12.2020]. Dostupné z: <https://www.doodly.com/>
- [18] RUSSELL, Joy A. *The Presenter's Toolbox* Journal of the Medical Library Association [online]. 2017, 01.07.2017 [cit. 28.03.2021]. Dostupné z: doi:10.5195/jmla.2017.210, (překlad vlastní)
- [19] SOLIRAX. *Neos VR* [software]. 2018, [cit 25.02.2021]. Dostupné z: <https://neos.com/>
- [20] KLÁN, Petr a Tomáš MARIANČÍK. *Jak stavět virtuální světy v metaverzu Neos: LogiX, avatary, Neos, virtuální realita*. [Litomyšl]: H.R.G. spol. s r.o., 2019. ISBN 978-80-88320-26-5.
- [21] KLÁN, Petr a Tomáš MARIANČÍK. Virtuální Přednáška VR11 – Vizuální programování LogiX 4/4 – doc. Petr Klán, CSc. – ČVUT. In: *Youtube* [online]. 13.12.2020 [cit. 25.02.2021]. Dostupné z: <https://www.youtube.com/watch?v=kKIxCxfHkeI&t=54s>. Kanál uživatele Frooxius
- [22] LEDDEN, Emma. *Úspěšná prezentace: získejte si své publikum krok za krokem*. Přeložil Eva MAŠKOVÁ KŘÁPKOVÁ. Praha: Grada, 2018. ISBN 978-80-247-2912-1.
- [23] VYNIKAROVÁ, Dana. *Dokumentace, prezentace a rétorika IX. přednáška – Zásady tvorby prezentace* [online prezentace]. 2020 [cit. 02.03.2021]. Dostupné z: https://moodle-vyuka.cvut.cz/pluginfile.php/307807/mod_resource/content/7/pr09_BI-DPRZS.pdf

-
- [24] LAURENČÍK, Marek. *Jak na dokonalou prezentaci v PowerPointu*. Praha: Grada, 2013. Snadno a rychle (Grada). ISBN 978-80-247-4377-6.
- [25] BARKLEY, Hudson. *Snu Slideshow Generator For Blender 2.83* [online]. 14.08.2020, [cit 03.12.2020]. Dostupné z: <https://github.com/snuq/SnuSlideshowGenerator>, (překlad vlastní)
- [26] HEMMER, Philipp (Hapit). *blender`addon`dynamic`slideshow* [online]. 17.04.2018, [cit 03.12.2020]. Dostupné z: https://github.com/hapit/blender_addon_dynamic_slideshow/wiki/Documentation, (překlad vlastní)
- [27] SKYWAYVISUALS. *NEW Dynamic Slideshow Template for Blender Skyway*. In: *YouTube* [online]. 15.09.2017, [cit 03.12.2020]. Dostupné z: <https://www.youtube.com/watch?v=HxiStxGi8gQ>, (překlad vlastní)
- [28] HIERHOLD, Emil. *Rétorika a prezentace: 7., aktualizované vydání*. Praha: Grada, 2008. Expert (Grada). ISBN 978-80-247-2423-2.
- [29] MICROSOFT CORP. *Get creative with 3D models* [online]. 23.06.2020 [cit. 04.03.2021]. Dostupné z: <https://support.microsoft.com/en-us/topic/get-creative-with-3d-models-ec5feb79-b0af-47f6-a885-151fcc88ac0a>, (překlad vlastní)
- [30] LOFTIN, Dorothy. *5 Ways 3D Models Can Help in Education* [online]. 15.08.2018 [cit. 04.03.2021]. Dostupné z: <https://graphicmama.com/blog/7-popular-software-presentations/>, (překlad vlastní)
- [31] VYNIKAROVÁ, Dana. *Dokumentace, prezentace a rétorika VIII. přednáška – Zásady prezentování* [online prezentace]. 2020 [cit. 10.03.2021]. Dostupné z: https://moodle-vyuka.cvut.cz/pluginfile.php/370501/mod_resource/content/1/pr08_BI-DPR.pdf
- [32] Opening .blend files using Blender's Python API. In: *Stack Overflow* [online]. 2015 [cit. 15.04.2021]. Dostupné z: <https://stackoverflow.com/questions/28075599/opening-blend-files-using-blenders-python-api>
- [33] BLENDER FOUNDATION. *Blender 2.92 Reference Manual – Library Overrides* [online]. 2021 [cit. 15.04.2021]. Dostupné z: https://docs.blender.org/manual/en/latest/files/linked_libraries/library_overrides.html
- [34] BLENDER FOUNDATION. *Blender 2.92 Reference Manual – Eevee* [online]. 2021 [cit. 18.04.2021]. Dostupné z: <https://docs.blender.org/manual/en/latest/render/eevee/introduction.html>, (překlad vlastní)

- [35] poll() failed, context incorrect?. In: *Blender Stack Exchange* [online]. 2019 [cit. 18.04.2021]. Dostupné z: <https://blender.stackexchange.com/questions/6101/poll-failed-context-incorrect-example-bpy-ops-view3d-background-image-add>
- [36] BLENDER FOUNDATION. *Blender 2.92 Reference Manual – Nonlinear Animation Strips* [online]. 2021 [cit. 18.04.2021] Dostupné z: <https://docs.blender.org/manual/en/latest/editors/nla/strips.html#action-strips>, (překlad vlastní)
- [37] ADOBE SYSTEMS INCORPORATED. *kernDump*. [online]. [cit. 20.04.2020]. Dostupné z: <https://github.com/adobe-type-tools/kern-dump>
- [38] WITT, Sebastian. *Add kerning support for 3DText* [online]. 08.08.2016 [cit. 20.04.2021]. Dostupné z: <https://developer.blender.org/D2282>
- [39] BARTON, Campbell. *Text objects ignore font kerning data* [online]. 03.09.2020 [cit. 20.04.2021]. Dostupné z: <https://developer.blender.org/T69805>, (překlad vlastní)
- [40] NUTTI. *Blender Add-on: Screencast Keys 3.5* [online]. 05.03.2021 [cit. 28.04.2021]. Dostupné z: <https://github.com/nutti/Screencast-Keys>
- [41] BLENDER FOUNDATION. *Blender 2.92 Reference Manual – Drivers* [online]. 2021 [cit. 05.05.2021] Dostupné z: <https://docs.blender.org/manual/en/latest/animation/drivers/introduction.html>, (překlad vlastní)
- [42] IT-SLOVNÍK.CZ. *Co je to Kerning?* [online]. 2021 [cit. 05.05.2021] Dostupné z: <https://it-slovník.cz/pojem/kerning>
- [43] BLENDER FOUNDATION. *Blender 2.92 Reference Manual – Markers* [online]. 2021 [cit. 05.05.2021] Dostupné z: <https://docs.blender.org/manual/en/latest/animation/markers.htm>, (překlad vlastní)
- [44] BLENDER FOUNDATION. *Classroom* [3D model]. [cit. 09.05.2021] Dostupné z: <https://www.blender.org/download/demo-files/>
- [45] ANDROMORA. *T-REX Dino TRex, Bones, Skeleton*. In: BlendSwap [3D model]. [cit. 09.05.2021] Dostupné z: <https://blendswap.com/blend/21854>
- [46] INGOENIUS. *Eiffel Tower*. In: BlendSwap [3D model]. [cit. 09.05.2021] Dostupné z: <https://blendswap.com/blend/8992>
- [47] MAAD, Ahmad. *Rome Colosseum*. In: BlendSwap [3D model]. [cit. 09.05.2021] Dostupné z: <https://blendswap.com/blend/3124>

- [48] MRCHIMP2313. *Soccer Stadium Large*. In: BlendSwap [3D model]. [cit. 09.05.2021] Dostupné z: <https://blendswap.com/blend/10567>

Slovník pojmů

drag-and-drop Operace, při které uživatel klikne levým tlačítkem myši na určitý objekt a zatímco tlačítko drží, posune myši a následně tlačítko pustí.

driver Pojem z prostředí aplikace Blender. Slouží k navázání určité akce na proměnnou, matematický vzorec, nebo funkci [41].

kerning Typografický pojem. Úprava mezer mezi určitými dvojicemi písmen (např. VA) [42].

layout Grafické rozvržení plochy.

marker Pojem z prostředí aplikace Blender. Slouží k označení určité události – začátek animace, změna kamery apod. [43].

placeholder Dočasný, či zástupný obsah, který bude v budoucnu nahrazen konkrétními daty.

Seznam použitých zkratk

HW Hardware

NLA Nonlinear Animation

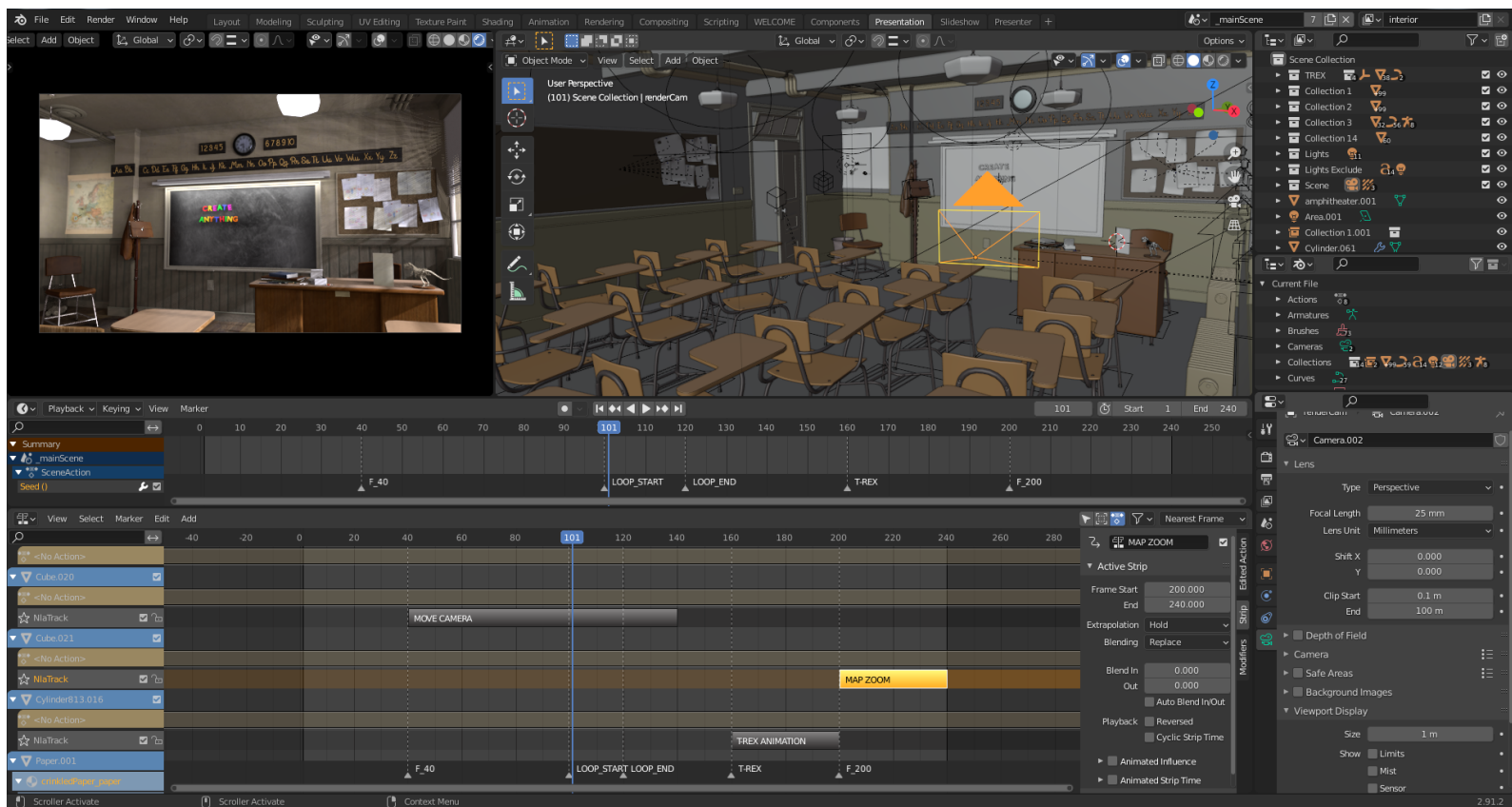
SW Software

UC Use-case, případ užití

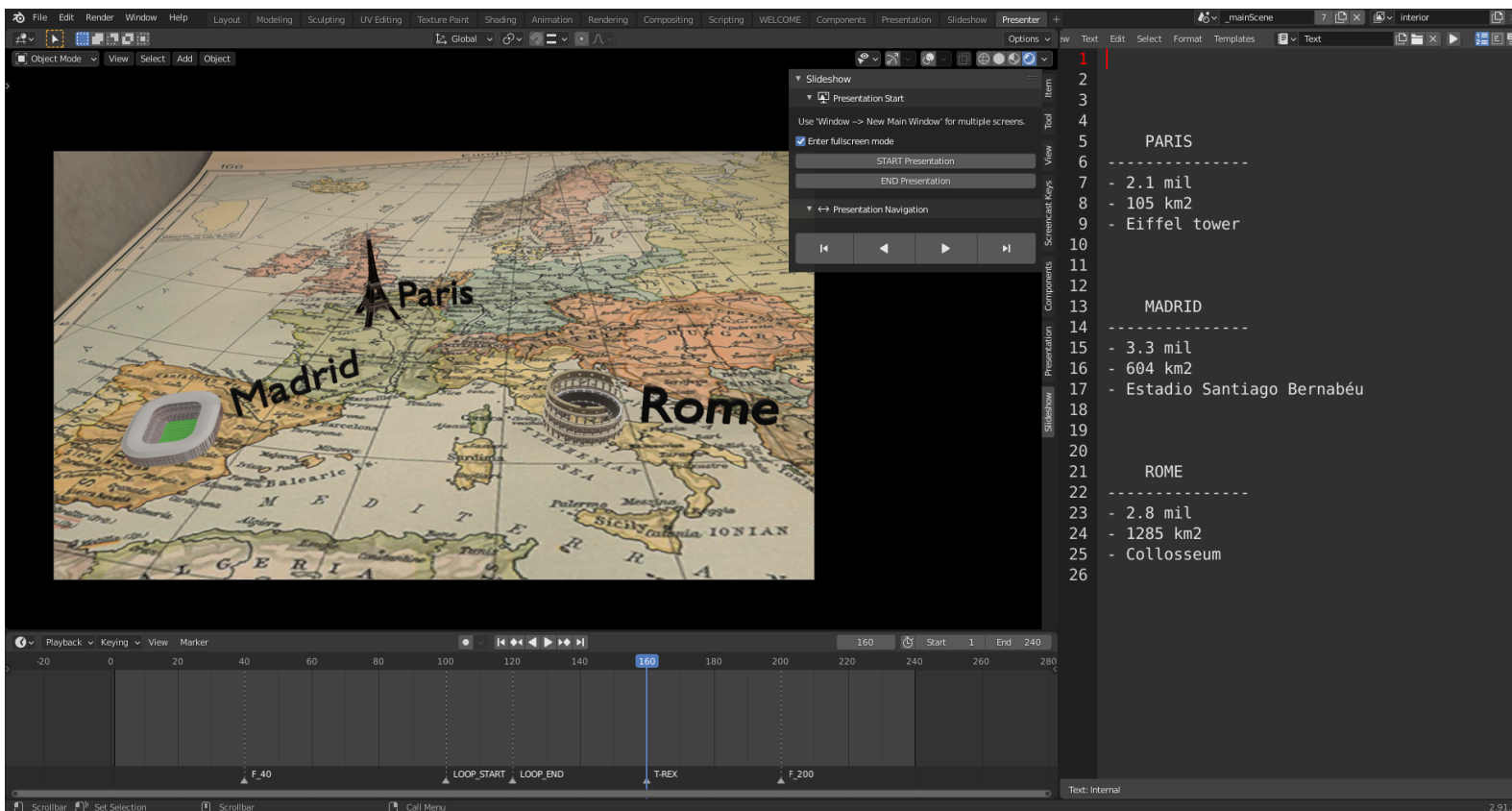
UI User interface

Ukázky z výsledného prezentačního systému

V ukázkách jsou použity 3D modely třídy [44], T-rexe [45], Eiffelovy veže [46], Kolosea [47] a fotbalového stadionu [48].



Obrázek C.1: Ukázka z workspace Presentation



Obrázek C.2: Ukázka z workspace Presenter



Obrázek C.3: Ukázka z workspace Slideshow

Obsah přiloženého CD

readme.txt	stručný popis obsahu CD
plugin_manual.txt	příručka pro instalaci a použití
src	
├── thesis	zdrojová forma práce ve formátu $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$
├── impl	
│ ├── presentation_plugin.py	výsledný plug-in
│ └── PRESENTATION_TEMPLATE.blend	šablona pro využití plug-inu
text	
└── thesis.pdf	text práce ve formátu PDF