



## Zadání bakalářské práce

<b>Název:</b>	Zásuvné moduly pro kreativní práce v programech Blender a Photoshop
<b>Student:</b>	Michal Rudolf
<b>Vedoucí:</b>	Ing. Jiří Chludil
<b>Studijní program:</b>	Informatika
<b>Obor / specializace:</b>	Webové a softwarové inženýrství, zaměření Počítačová grafika
<b>Katedra:</b>	Katedra softwarového inženýrství
<b>Platnost zadání:</b>	do konce letního semestru 2023/2024

### Pokyny pro vypracování

Cílem této práce je vytvoření prototypových zásuvných modulů pro komunitu umělců pracujících v prostředí programů Blender a Adobe Photoshop.

1. Analyzujte postupy práce členů umělecké komunity a definujte jejich požadavky na moduly.
2. Analyzujte stávající řešení (import/export) kompozičních vrstev mezi programy.
3. Pomocí metod softwarového inženýrství navrhnete prototypy zásuvných modulů řešící danou problematiku.
4. Navržené řešení implementujte.
5. Výsledné moduly podrobte uživatelským testům.



Bakalářská práce

**ZÁSUVNÉ MODULY  
PRO KREATIVNÍ PRÁCE  
V PROGRAMECH  
BLENDER  
A PHOTOSHOP**

**Michal Rudolf**

Fakulta informačních technologií  
Katedra softwarového inženýrství  
Vedoucí: Ing. Jiří Chludil  
2. ledna 2023

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2023 Michal Rudolf. 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, je nezbytný souhlas autora.*

Odkaz na tuto práci: Rudolf Michal. *Zásuvné moduly pro kreativní práce v programech Blender*

*a Photoshop*. Bakalářská práce. České vysoké učení technické v Praze, Fakulta informačních technologií, 2023.



# Obsah

Poděkování	vi
Prohlášení	vii
Abstrakt	viii
<b>1 Úvod do problematiky</b>	<b>1</b>
1.1 Digitální malba a renderování	1
1.2 Zásuvné moduly pro grafické aplikace	1
1.2.1 Photoshop	2
1.2.2 Blender	2
1.3 Pracovní postup Blender-Photoshop	3
<b>2 Průzkum</b>	<b>7</b>
2.1 Cílová skupina	7
2.2 Postup průzkumu	7
2.2.1 Otázka č. 1 – Pracovní postup	8
2.2.2 Otázka č. 2 – Úskalí pracovního postupu	8
2.2.3 Otázka č. 3 – Používané zásuvné moduly	8
2.2.4 Otázka č. 4 – Kompoziční vrstvy a OpenEXR	8
2.2.5 Otázka č. 5 – Opětovné renderování	9
2.3 Zkušenosti s prováděním průzkumu	9
2.4 Výsledky průzkumu	9
2.4.1 Rozbor odpovědí – Pracovní postup	9
2.4.2 Rozbor odpovědí – Úskalí pracovního postupu	10
2.4.3 Rozbor odpovědí – Používané zásuvné moduly	10
2.4.4 Rozbor odpovědí – Kompoziční vrstvy a OpenEXR	11
2.4.5 Rozbor odpovědí – Opětovné renderování	11
<b>3 Analýza</b>	<b>13</b>
3.1 Programování s knihovnou Blender Python API	13
3.2 Výběr způsobu tvorby modulu pro Photoshop	14
3.3 Příklady současných řešení	15
3.3.1 Photoshop	15
3.3.2 Blender	16
3.4 Problémy pracovního postupu Blender-Photoshop	17
3.5 Funkční a nefunkční požadavky	19
<b>4 Návrh</b>	<b>21</b>
4.1 Zásuvný modul pro Blender	21
4.2 Zásuvný modul pro Photoshop	27

<b>5 Implementace</b>	<b>29</b>
5.1 Schéma nasazení . . . . .	29
5.2 Instalační příručka . . . . .	30
5.3 Uživatelská příručka . . . . .	31
5.4 Programátorský příručka . . . . .	33
5.5 Komplikace a jejich řešení . . . . .	34
5.6 Kontrola splnění funkčních požadavků . . . . .	35
<b>6 Testování</b>	<b>37</b>
6.1 Scénář uživatelského testu pro umělce . . . . .	37
6.2 Výsledky testů . . . . .	38
6.3 Prostor pro zlepšení . . . . .	38
<b>7 Závěr</b>	<b>39</b>
<b>Obsah přiloženého média</b>	<b>43</b>

## Seznam obrázků

1.1	Roční návštěvnost webu blender.org . . . . .	3
1.2	Pracovní postup Blender-Photoshop . . . . .	3
1.3	Příklady kompozičních vrstev . . . . .	5
1.4	Clown pass - Autor: Shawn Ma [11] . . . . .	6
3.1	Nastavení modulu EXR-IO při importu . . . . .	16
4.1	Diagram aktivit zásuvného modulu pro Blender . . . . .	22
4.2	Návrh zapojení kompozičních uzlů v Blenderu . . . . .	24
4.3	Kompoziční uzly . . . . .	25
4.4	Návrh uživatelského rozhraní pro Blender modul . . . . .	26
4.5	Diagram aktivit zásuvného modulu pro Photoshop . . . . .	27
5.1	Uživatelský postup práce s moduly . . . . .	29
5.2	Instalace modulu v aplikaci Blender (verze 3.1.2) . . . . .	30
5.3	Uložení skriptu do Presets . . . . .	31
5.4	Volání skriptu (PS verze 24.1.0) . . . . .	31
5.5	Uživatelský postup práce s moduly . . . . .	32
5.6	Uživatelské rozhraní modulu pro Blender . . . . .	32

## Seznam tabulek

1.1	Často používané Kompoziční vrstvy . . . . .	4
5.1	Splnění požadavků . . . . .	35

## Seznam výpisů kódu

3.1	Výpis kódu č. 1 (Python) . . . . .	14
5.1	Registrace modalu . . . . .	34

*Chtěl bych poděkovat především panu Ing. Chludilovi za vedení mé práce. Dále bych chtěl poděkovat všem účastníkům mého průzkumu, za to, že si našli čas a zamysleli se nad mými otázkami. V neposlední řadě bych chtěl poděkovat své rodině a přátelům, kteří mi nabídli rady a psychickou podporu při psaní.*

## 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 2. ledna 2023

.....

## Abstrakt

Kombinace postupů s grafickými aplikacemi Blender a Photoshop se používá v oborech digitálního umění. Problémem je, že tyto postupy obsahují technicky složité úkoly a repetitivní kroky. Na toto téma jsem provedl průzkum, ve kterém jsem s profesionály v oborech digitální malby diskutoval o jejich každodenních pracovních postupech. Na základě výpovědí o jejich zkušenostech s programy jsem vytvořil sadu zásuvných modulů, které zrychlují a zjednodušují přesun kompozičních dat mezi programy Blender a Photoshop. Moduly také nabízí nový způsob generování pixelových masek, které pak výrazně ulehčují práci v následujících fázích malby. Výsledné moduly jsem podrobil uživatelským testům, které vykonali účastníci průzkumu.

**Klíčová slova** Blender, Photoshop, zásuvné moduly, optimalizace pracovního postupu, import kompozičních vrstev, export kompozičních vrstev

## Abstract

Combination of workflows with graphic softwares Blender and Photoshop is used in digital art industry. The problem is, that these workflows contain technically difficult and repetitive tasks. On this topic, I have conducted a survey, in which I discussed with profesional artists about their workflows. Based on insight from the responses I have made a set of plug-ins, that speed up and simplify transferring of composition data between Blender and Photoshop. Apart from that, they also provide better solution for mask generation, which is benefitial in later stages. Plug-ins are tested by members of the survey.

**Keywords** Blender, Photoshop, plug-ins, workflow optimalization, render pass import, render pass export

## Seznam zkratek

API	Application Programming Interface
CC	Creative Cloud
CEP	Common Extensibility Platform
CS	Creative Suite
DLL	Dynamic-link library
GPL	General Public License
JSDoc	Java Script Documentation
PEP	Python Enhancement Proposals
PS	Photoshop
SDK	Software development kit
UXP	Unified Extensibility Platform
VFX	Visual effects





# Úvod

V zábavních průmyslech, které produkují vizuální vjemy, se často využívají kreativní služby designérů a umělců. Jedná se o velmi širokou škálu kreativních prací s rozmanitostí klientských požadavků. Díky tomu existuje velké množství grafických aplikací a žádný jednotný postup. Většina umělců si vytváří vlastní pracovní postup, aby dosáhli požadované kvality a jejich postupy často zahrnují práci s více programy.

V této práci se zaměřuji na postupy kreativních prací zabývajících se tvorbou 2D rastrové grafiky, konkrétně digitální malbu. V digitální malbě jsou často využívány grafické programy Blender a Photoshop, které podporují tvorbu zásuvných modulů. Mým cílem je vytvořit pro tyto grafické aplikace prototypy zásuvných modulů, které budou zjednodušovat přesun dat mezi programy a tím umělecké komunitě šetřit čas a úsilí.

Vybral jsem si digitální malbu ze dvou důvodů. Zaprvé, jedná se o disciplínu, která se využívá napříč odvětvími zábavních průmyslů a zadruhé, mám s touto disciplínou dlouholeté osobní zkušenosti a mám v ní dostatek znalostí, abych mohl s profesionály vést konverzaci o jejich pracovních postupech. Téma jsem si vybral, abych se zásuvnými moduly odvděčil umělecké komunitě za její otevřenost a pomoc začínajícím umělcům jako jsem byl já.

Tato práce je určena zájemcům, kteří chtějí vědět, jak funguje tvorba zásuvných modulů pro grafické aplikace nebo tvorba zásuvných modulů pro zlepšení produktivity obecně.

V teoretické části začínám obecným úvodem do problematiky zásuvných modulů a vysvětlením některých specifických mechanik, které se používají v digitální malbě. Následuji průzkumem mezi profesionály, kteří ke své práci využívají právě programy Blender a Photoshop. Průzkumu nabídne čtenáři pohled do pracovních postupů lidí, kteří mají v oboru digitální malby rozsáhlé zkušenosti. Jako další provádím analýzu odpovědí, ve které hledám problémy pracovního postupu Blender-Photoshop, které lze řešit zásuvnými moduly. Následně popisuji příklady stávajících řešení, která přispívají komunitě umělců po celém světě. Na konec teoretické části sestavuji funkční požadavky pro moduly.

V praktické části vytvářím návrh prototypů zásuvných modulů s pomocí metod softwarového inženýrství. Dále popisuji postup implementace prototypů a stručně vysvětluji, jak s moduly pracovat. Na závěr prototypy podrobím uživatelským testům s pomocí respondentů z průzkumu a zhodnotím přínos vytvořených zásuvných modulů umělecké komunitě.



# Cíle

Cílem této bakalářské práce je vytvořit prototypy zásuvných modulů, které budou zlepšovat efektivitu práce s grafickými aplikacemi Blender a Photoshop. Prototypy budou sloužit k optimalizaci repetitivních úkolů v každodenním pracovním postupu umělců, konkrétně se budou zásuvné moduly zabývat přesunem kompozičních dat mezi programy.

Cílem teoretické části provést průzkum v umělecké komunitě, určit společné pracovní postupy, najít úskalí pracovních postupů a vysvětlit jejich specifické mechaniky. Dalším cílem je prozkoumat stávající řešení zásuvných modulů pro grafické aplikace a zhodnotit jejich přínos umělecké komunitě. Po analýze výsledků průzkumu je cílem definovat funkční požadavky pro zásuvné moduly.

Cílem praktické části je navrhnout prototypy zásuvných modulů pomocí metod softwarového inženýrství a navržené prototypy implementovat. Dalším cílem je pak podrobit zásuvné moduly uživatelským testům a zhodnotit jejich přínos umělecké komunitě.



# Úvod do problematiky

*V následující kapitole vysvětluji postupy a pojmy specifické pro tvorbu zásuvných modulů a pro digitální malbu v programech Blender a Photoshop.*

## 1.1 Digitální malba a renderování

Digitální malba je forma umění využívající potenciál digitálních nástrojů dnešních počítačů. Malba v digitálním prostředí nabízí řadu výhod. Nejdůležitější výhodou je pohodlnost: digitální díla je jednoduché přesouvat, sdílet, jednoduché tisknout a opravovat. Díky tomu stále více umělců přesouvá svá portfolia do nehmotného světa počítačů [1]. Digitální malba samozřejmě zahrnuje nespočetná umělecká odvětví s různými postupy práce a různými výstupy.

V mé práci se budu soustředit na procesy s výstupy 2D rastrové grafiky, které vznikly renderováním 3D scény v programu Blender a následně digitální malbou přes vyrenderovaný výstup v programu Photoshop. Tento postup kombinuje sílu fyzikálních modelů render engine s kreativitou umělců. Digitální díla se pak využívají v produktech zábavních průmyslů, zejména v rané fázi tvorby filmů a her. Někteří umělci využívají více různých 3D softwarů v procesu vytváření díla, aby tak zrychlili jejich pracovní postup [2]. Umělci, zabývající se převážně 3D designem, obvykle používají Photoshop na post-processing (dodatečné úpravy výstupu renderování) [3]. Na druhou stranu umělci, kteří se specializují spíše na 2D ilustrace, používají 3D software, jako je Blender, k tvorbě perspektivních nebo kompozičních základů pro svá díla. Poměr úsilí jaké umělec vynaloží na 3D versus 2D je velmi individuální [4].

## 1.2 Zásuvné moduly pro grafické aplikace

V konceptním umění s výstupem 2D grafiky se často využívají programy ZBrush, Modo, Maya, Painter, Sketchbook Pro, Blender a Photoshop[5]. V této práci se zaměřuji na aplikace Blender a Photoshop. Jsou to samostatné aplikace s vlastními vývojovými góly, jejich kombinace při tvorbě ilustrací je pouze jeden z mnoha uživatelských postupů. Mezi programy není intuitivní přesun dat a uživatelé jsou nuceni přidávat do svých pracovních postupů technicky složitě a repetitivní úkoly. To jsou přesně místa, která se hodí vyplnit zásuvnými moduly.

Zásuvný modul (plug-in, add-on) je krátký program, který pracuje jako doplňkový modul jiné aplikace a rozšiřuje tak její funkcionalitu. Photoshop i Blender podporují externí vývoj zásuvných modulů tím, že vydají ke svému programu API knihovnu a sadu pravidel jak s ní pracovat. Vývojáři zásuvných modulů implementují své řešení a s pomocí API knihovny jej spojí s rozhraním aplikace. Výsledkem je malý program ovladatelný přímo z nativního prostředí aplikace.

Zásuvné moduly můžou sloužit různým účelům jako například: přidat novou funkcionalitu, vylepšit stávající funkcionalitu automatizovat/optimalizovat/zjednodušit sadu kroků, nabídnout náměty pro kreativitu, umožnit využití dosud nepodporovaného formátu, zpřehlednit nebo zjednodušit složité uživatelské rozhraní programu a mnoho dalších. Jejich využití je limitováno pouze představivostí a dovednostmi autora a možnostmi API knihovny programu.

### 1.2.1 Photoshop

Photoshop je již mnoho let synonymem pro úpravu rastrových obrázků. I po tom, co přešel na měsíční platební plán, proslulý software zůstává předním řešením pro umělce, ilustrátory, designéry a fotografy. Jeho nástroje na vrstvy, masky a retuš jsou stále standardem, podle kterého se soudí jiné programy, zatím co síla systému založeného na vrstvách dokáže zvládnout ty nejkomplexnější kreativní projekty [6].

Aplikace Photoshop společně s grafickým tabletem nabízí prostředí pro kresbu hrubého návrhu, malbu obrázku i finální úpravy. Užitečné pro umělce jsou zejména nástroje pro tvorbu vlastních štětců, pixelových masek a barevných přechodů. Tyto nástroje, a mnoho dalších, pomáhají předat myšlenku na digitální plátno rychleji a snadněji, než bylo možné v tradičním umění a proto většina umělců zahrnuje Photoshop ve svém pracovním procesu.

#### Skripty

Možnost tvořit skripty pro Photoshop v jazyce JavaScript začala být podporována ve verzi 7 jako volitelný zásuvný modul stáhnutelný z oficiální stránky Adobe. Ve verzi 8 (první verze Creative Suite) byla podpora skriptů instalována jako výchozí nastavení. I přes to, že je skriptování silným nástrojem Photoshopu, velmi málo lidí ví, že je k dispozici, a ještě méně ho využívá [7].

#### Photoshop SDK

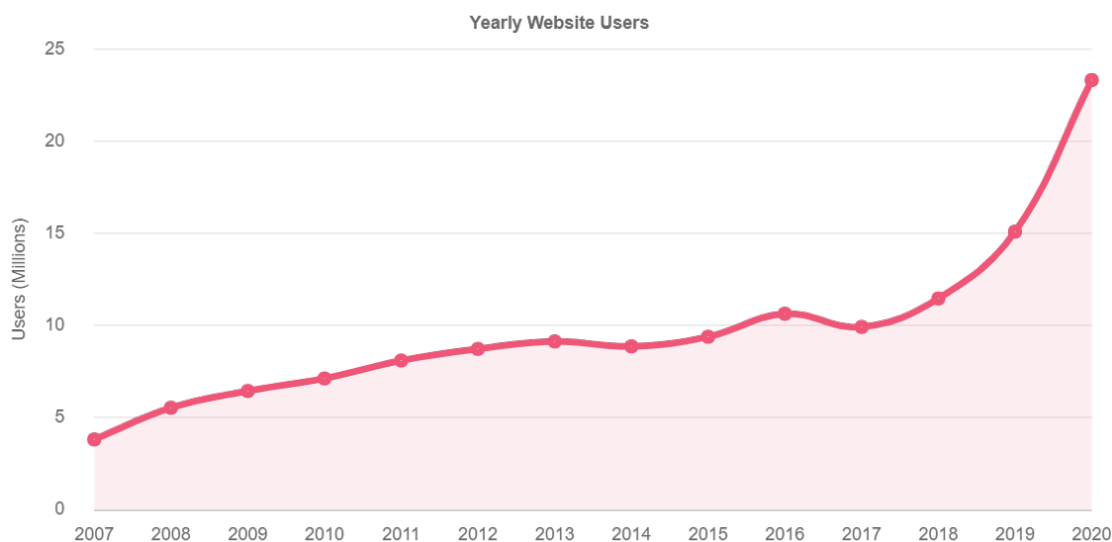
Společnost Adobe Inc. navíc podporuje vývoj externích zásuvných modulů pro většinu svých aplikací. Pro Photoshop se v průběhu let vývoj API knihoven rozvětvil, a tak dnes máme více knihoven, které umožňují tvorbu zásuvných modulů. Historicky první, stále podporovaná v nejnovějších verzích aplikace, je Adobe Photoshop SDK. Jedná se o knihovnu napsanou v jazyce C++. Společnost Adobe Inc. udržuje podporu této knihovny, ale od roku 2020 nepokračuje s jejím vývojem. Velkou nevýhodou této knihovny je špatná dokumentace a složitá kompilace.

#### Photoshop UXP API

Druhou knihovnou je Photoshop UXP API, která byla vydána v roce 2020. Na rozdíl od Adobe Photoshop SDK, zásuvné moduly s využitím Photoshop UXP API se programují s pomocí moderních webových technologií HTML, CSS a JavaScript. Použití Photoshop UXP API umožňuje také nabízet zásuvný modul na online tržišti Adobe Marketplace, přístupné všem se zakoupenou licencí Adobe Creative Cloud. Knihovna Photoshop UXP API patří do skupiny knihoven Adobe CEP (Common Extensibility Platform), která umožňuje vyvíjet moduly podporované více aplikacemi od Adobe. Jedná se o společné prostředí pro vývoj modulů, a tak lze vyvinout modul, který bude použitelný v aplikaci Photoshop i Ilustrátor zároveň.

### 1.2.2 Blender

Blender za poslední roky získala velkou popularitu. V přelomových letech 2018 a 2019 se mu podstatně zvýšila návštěvnost hlavní webové stránky (viz. obrázek č. 1.1) a i počet vývojářů, kteří přispívají k open-source projektu [8]. Blender začíná konkurovat hollywoodským 3D softwarům jako například Cinema4D nebo 3D Studio MAX, avšak narozdíl od nich je zcela zadarmo a jeho



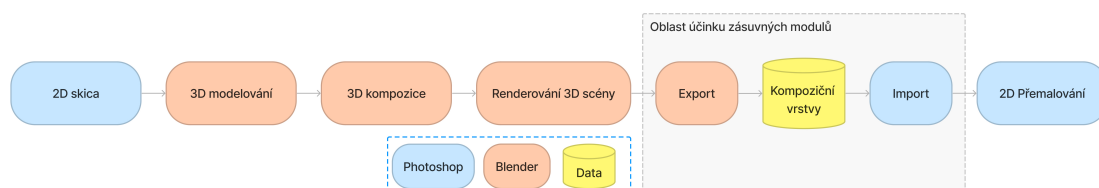
■ **Obrázek 1.1** Roční návštěvnost webu blender.org

zdrojové kódy jsou volně k dispozici pod licencí GNU GPL. Podle článku z magazínu Trip Wire dokonce patří ke dvaceti nejpopulárnějším open-source projektům všech dob [9].

## Blender Python API

Blender podporuje tvorbu zásuvných modulů v jazyce Python knihovnou Blender Python API. Dokumentace Blender Python API je na vysoké úrovni a obsahuje dokonce i variaci přednastavených šablon, které lze při vývoji využít. Na internetu existuje velké množství manuálů a příkladů, které pomáhají nováčkům pochopit složité koncepty. Vydavatelé Blenderu nabízí sadu přednastavených zásuvných modulů, které lze zapnout přímo v aplikaci. Zásuvné moduly vytvořené externími vývojáři lze nainstalovat a spustit v aplikaci. Veškeré zásuvné moduly mají formu python skriptu, který nepožaduje žádnou kompilaci.

## 1.3 Pracovní postup Blender-Photoshop



■ **Obrázek 1.2** Pracovní postup Blender-Photoshop

Na obrázku č.1.2 jsem označil, jak zhruba vypadá postup prací s využitím grafických programů Blender a Photoshop. Zároveň jsem vyznačil oblast, na kterou se zaměřím při tvorbě zásuvných modulů v praktické části. Tuto oblast jsem si vybral, protože i když se pracovní postupy umělců často liší, export dat z Blenderu a import do Photoshopu řeší každý uživatel, takže zásuvné moduly, které v této práci budu vytvářet budou mít větší cílovou skupinu, a tak i větší přínos komunitě.

■ **Tabulka 1.1** Často používané Kompoziční vrstvy

Krátký název	Popis
Combined	Finální render (kombinovaná vrstva)
Z	Zobrazení hodnot hloubky objektů
Mist	Zobrazení hloubky scény ve formě mlhy - užitečné pro maskování
Normal	Zobrazení normál - užitečné například při malbě sněhu
DiffDir	Zobrazení direktního difuzního světla
DiffInd	Zobrazení odraženého difuzního světla
DiffCol	Zobrazení barvy difuzního světla
GlossDir	Zobrazení direktního spekulárního světla
GlossInd	Zobrazení odraženého spekulárního světla
GlossCol	Zobrazení barvy spekulárního světla
Emit	Zobrazení záření ve scéně
Denoise Data	Data z odšumění kombinované vrstvy - k odšumění kompozičních vrstev
Shadow	Zobrazení všech stínů ve scéně
Clown	Zobrazení každého objektu s jinou barvou - užitečné pro selekci

Práce s oběma programy dokáže být velmi složitá. Pracovní postup využívající Blender a Photoshop obsahuje některé specifické procesy a termíny, které je třeba vysvětlit.

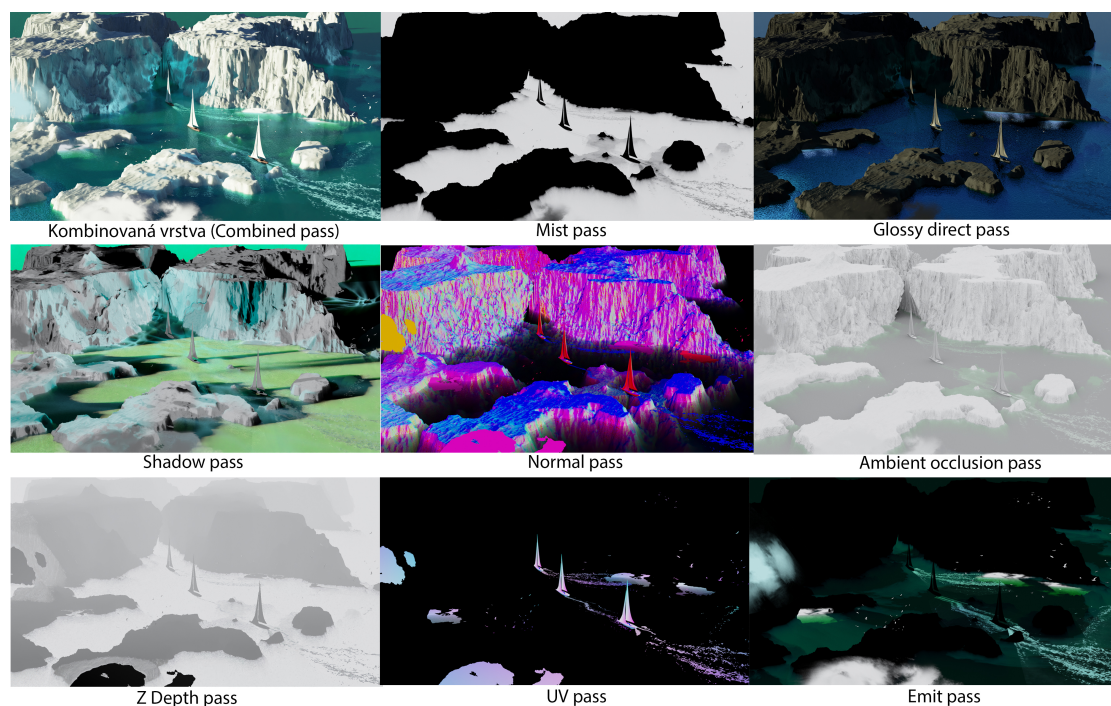
## Kompozitor

Okno Kompozitoru najdeme v aplikaci Blender. Je to prostor, ve kterém lze s pomocí vizuálního programování upravovat výstupy renderovacího procesu. Vizuální programování v Kompozitoru umožňuje volat funkce nad zvolenými výstupy bez psaní kódu. Místo toho uživatel vybírá ze sbírky kompozičních uzlů a spojuje jejich vstupy a výstupy s ostatními uzly. Vytvářením uzlů a spojení mezi nimi uživatel vytváří sadu instrukcí, kterou pak Blender spustí po dokončení renderovacího procesu. V některých uzlech lze nastavit doplňkové vlastnosti funkce, kterou reprezentuje. Tyto vlastnosti lze nastavit v uživatelském rozhraní, v příkazové řádce nebo ze zásuvného modulu. Kompozitor je jedním z kandidátů na umístění zásuvného modulu, protože kompoziční uzly lze generovat a nastavovat příkazy.

## Kompoziční vrstvy

Kompoziční vrstvy (render pass) jsou jednou z výhod pracovního postupu Blender-Photoshop. Kompoziční vrstvy jsou disjunktní vrstvy výstupu render engineů (ve výchozím nastavení Blenderu je podporují enginy Cycles a Eevee), které je možné vygenerovat společně s kombinovanou vrstvou. Kombinovaná vrstva (combined pass) je nejdůležitější vrstvou, protože to je finální obrázek, který je výstupem renderování. Často používané kompoziční vrstvy a jejich účel jsem popsal v tabulce č.1.1. Pokud tyto vrstvy importuji do Photoshopu, tak získám volnost upravovat každou zvlášť. To nejen výrazně šetří čas při úpravě, ale i umožňuje do jisté míry měnit vlastnosti scény po vyrenderování ve fázi 2D přemalování (viz. obrázek č.1.2). Na obrázku č.1.3 lze vidět jak se od sebe jednotlivé vrstvy liší. Kompoziční vrstvy, jako například Normálová vrstva (Normal pass) nebo Hloubková vrstva (Z pass), obsahují informace, ze kterých se neskládá kombinovaná vrstva, ale mohou být užitečné při úpravě ve Photoshopu. Naopak vrstvy jako třeba vrstva direktního difuzního světla (DiffDir pass) nebo vrstva záření (Emit pass) přímo tvoří složky, ze kterých je vytvořena kombinovaná vrstva.





■ **Obrázek 1.3** Příklady kompozičních vrstev

## Render vrstvy

Aplikace Blender nabízí pro každou scénu vytvářet render vrstvy. Tyto vrstvy mají identický obsah, ale uživatel může, kromě dalších nastavení, jednotlivé objekty schovat. Tímto lze vyrenderovat pozadí objektů, které budou v budoucnu podléhat změně pozice. Každá render vrstva má svojí kolekci nastavení, ve které lze povolit nebo zakázat renderování vrstvy. Renderování více vrstev však značně prodlužuje renderovací čas.

## Formát OpenEXR

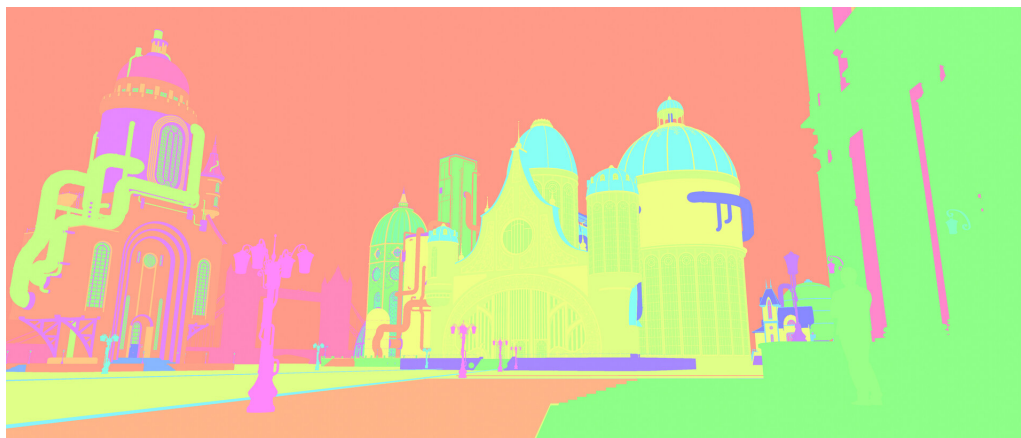
OpenEXR je souborový formát, který umožňuje seskupení více souborů s obrázkovými daty do jednoho souboru. Interně formát nabízí ukládání v různých velikostech na pixel (až 32-bit) s možností ztrátové i bezztrátové komprese. Tento formát se často používá při tvorbě vizuálních efektů VFX, protože se s ním snadno manipuluje[10]. Aplikace Blender nabízí OpenEXR jako jeden z výstupních formátů a zároveň aplikace Photoshop obsahuje importační nástroj pro tento formát, takže má smysl tuto variantu prozkoumat.

## Pixelové masky

Maskování objektů je esenciální krok v procesu většiny umělců. Díky pixelovým maskám se může umělec pracovat rychle s různě velkými štětci bez starosti, že by štětcem přetáhl oblast objektu. Veškeré změny lze aplikovat pouze na oblast specifikovanou pixelovou maskou.

Pokud uživatel pracuje pouze v programu Photoshop, běžný postup tvorby masek pro objekty je, vzít si nástroj 'laso' a ručně vybrat plochu objektu. To je velmi časově náročný proces a jeho náročnost se násobí s komplexitou scény.

Pracovní postup Blender-Photoshop umožňuje tento proces automatizovat více způsoby. První způsob je render kompoziční vrstvy s plochými barvami objektů (tzv. „clown pass“). Na



■ **Obrázek 1.4** Clown pass - Autor: Shawn Ma [11]

vrstvě by dva objekty na scéně, které spolu vizuálně sousedí neměly mít stejnou barvu. Ve Photoshopu pak lze nástrojem Magická hůlka automaticky selektovat spojitou plochu určité barvy a vytvořit tak výběr, který slouží při malování štětcem jako maska objektu.

Dalším způsobem je generování kompozičních vrstev s označením 'cryptomatte'. Tyto vrstvy lze využít pro maskování jednotlivých objektů nebo i materiálů. Pokud uživatel povolí tvorbu cryptomatte dat, každý objekt si vygeneruje identifikátor. Poté při renderování lze s pomocí identifikátorů vyříznout plochu obrázku, kterou pokrývá daný objekt. Výhodou je, že výstupní vrstvy s aplikovanou maskou jsou vyříznuté z finálního renderu, a proto jsou již automaticky odšuměné. Výsledkem je zvolený objekt vyříznutý na separátní vrstvě s transparentním pozadím. S pomocí těchto separátních vrstev lze ve Photoshopu jednoduše uzamykat oblast úpravy nebo přesouvat objekt po plátně.

## Kapitola 2

# Průzkum

*V následující kapitole provedu čtenáře průzkumem, který jsem provedl, abych byl schopný identifikovat úskalí v pracovních postupech cílové skupiny.*

### 2.1 Cílová skupina

Jako cílovou skupinu jsem si vybral komunitu umělců pracujících s programy Blender a Photoshop. Jedná se o umělce, jejichž typickým zaměstnáním je tvorba konceptního umění a ilustrace. Konceptní umění i ilustrace se využívají při tvorbě filmů, her, komiksů, knih nebo animací. Tuto cílovou skupinu jsem si vybral, protože s jejich pracovními postupy mám osobní zkušenost.

### Platforma Artstation

Výhoda mé volby cílové skupiny spočívá v tom, že pro tyto umělce je profesním standardem mít účet a sdílet své portfolio na platformě Artstation. Když umělec sdílí dílo, má možnost přidat informaci, které programy ke tvorbě použil a z průzkumu je jasné vidět, že Blender-Photoshop je velmi používaný postup mezi začátečníky i profesionály v oboru.

Platforma Artstation je hlavním střediskem umělecké komunity. Slouží jako sociální síť, tržiště, výuková platforma, blogovací platforma, platforma pro nabídku/poptávku zaměstnání v oboru. Členové si zde mohou vytvořit vlastní portfolio, procházet portfolia ostatních členů, prozkoumávat celou databázi. V neposlední řadě členové platformy mohou navázat komunikaci přes integrovaný instant-messaging systém, který zasílá notifikaci příjemci zprávy na jeho osobní e-mail.

Rozhodl jsem se stát se členem a využít tuto platformu pro můj průzkum.

### 2.2 Postup průzkumu

K tomu být schopný zvolit správnou strategii pro optimalizaci je třeba provést důkladný průzkum. Cílem zásuvného modulu pro zlepšení produktivity je šetřit čas. Aby modul splnil tento cíl, musí jej členové cílové skupiny považovat za užitečný. Teprve tehdy se stane součástí pracovního postupu a bude mít přínos pro cílovou skupinu.

V průběhu korespondence jsem vymýšlel nebo z odpovědí respondentů přebíral zajímavé nápady na optimalizace. Tyto nápady jsem pak navrhnul ostatním respondentům s otázkou, zda by to pomohlo zefektivnit jejich pracovní proces. Tyto otázky už byly více specifické a dotazoval jsem se vždy většiny aktivních respondentů.

Ještě před zahájením průzkumu je třeba sdělit potencionálnímu respondentovi kontext, ze kterého vycházím, proto jsem se vždy představil a popsal účely mého průzkumu.

### 2.2.1 Otázka č. 1 – Pracovní postup

Při rozesílání prvních zpráv jsem si vybíral umělce, kteří specifikovali na serveru Artstation, že pracovali s programy Blender a Adobe Photoshop, abych měl jistotu, že spadají do cílové skupiny. První otázkou průzkumu jsem zamýšlel zjistit, jaké pracovní postupy cílová skupina využívá. Pracovní postupy jednotlivců se samozřejmě liší. Závisí na individuální úrovni dovedností, ať už kreativních, tak technických. Někteří umělci využívají více modelování ve 3D softwaru, jiní vytváří jen hrubé modely a většinu práce tvoří ve 2D softwaru. Pokud však mám dostatečně velký vzorek respondentů, mohu najít průnik jejich postupů a tyto společné postupy následně podrobit detailní analýze.

- Otázka č. 1: Všiml jsem si, že při práci používáte Blender a Photoshop, jaký používáte postup při své práci?

### 2.2.2 Otázka č. 2 – Úskalí pracovního postupu

Druhá otázka dává respondentům možnost popsat jejich negativní zkušenosti s pracovním postupem Blender-Photoshop. Dle mého názoru, je to nejdůležitější část průzkumu. Nechat je zamyslet se nad tím, jaké úkoly v jejich pracovním postupu dělají neradi nebo jim zabírají zbytečně moc času.

Nechat respondentům prostor přijít s vlastními nápady na optimalizace mi zároveň pomáhá při výběru potencionálních členů uživatelského testování. Pokud prototypy vytvořené pro tuto práci budou popsané úskalí optimalizovat, je větší pravděpodobnost, že se bude respondent účastnit testování.

- Otázka č. 2: Jaká jsou úskalí vašeho pracovního procesu Blender-Photoshop? Jaké jsou časově nejnáročnější nekreativní úkoly, které berou čas a nadšení ze zábavného kreativního procesu?

### 2.2.3 Otázka č. 3 – Používané zásuvné moduly

Další otázkou průzkumu měla za úkol zjistit, zda respondenti již využívají některé zásuvné moduly. Velká část respondentů nemá bohaté technické dovednosti nebo vůbec nad produktivitou svých pracovních postupů nepřemýšlí. Naopak jiní tomu věnují značné úsilí a sdílejí nadšení pro zlepšení produktivity. Tímto krokem lze zjistit, které úskalí pracovního postupu již byly adresovány nebo jaké náležitosti musí zásuvný modul mít, aby s ním uměl pracovat umělec s průměrnými technickými dovednostmi.

- Otázka č. 3: Využíváte ve svém pracovním procesu některé zásuvné moduly pro Blender nebo pro Photoshop?

### 2.2.4 Otázka č. 4 – Kompoziční vrstvy a OpenEXR

Čtvrtá otázka se zaměřuje na známé úskalí v jakémkoliv pracovním procesu, ve kterém se sériově využívá více programů; Přesun dat mezi programy. Cílem bylo zjistit jaké kompoziční vrstvy respondenti používají a jestli ví o možnosti exportování kompozičních vrstev ve formátu OpenEXR.

- Otázka č. 4: Používáte různé kompoziční vrstvy ve svém pracovním procesu? Pokud ano, exportujete je společně ve formátu OpenEXR?

## 2.2.5 Otázka č. 5 – Opětovné renderování

Pátá otázka se věnuje problému opětovného renderování scény. Umělci vytvářející koncepty v průběhu práce komunikují s klienty. Pokud klient požaduje malou či velkou změnu, většinou to vyústí v opětovné renderování.

- Otázka č. 5: Jak často narazíte na problém, že musíte znovu vyrenderovat celou 3D scénu? Zabírá vám tento aspekt pracovního procesu Blender-Photoshop hodně času?

## 2.3 Zkušenosti s prováděním průzkumu

Prvním krokem bylo vytvořit sadu otázek, jejichž odpovědi pomohou analýze problematiky. Pro tento krok jsem si vytvořil digitální formulář s tím, že budu rozesílat odkaz na něj a formulář povede dialog s respondenty za mě.

Po krátké zkušební fázi jsem zjistil, že tento způsob nevede k úspěšným výsledkům při navázání kontaktu. Důvody jsem vypožoroval dva: Prvním důvodem je, že dialog přes formulář nebere ohled na již zodpovězené otázky a tím pádem nedává možnost na reakci a na žádost o detailnější vysvětlení nebo rozvinutí odpovědi. Druhý důvod je subjektivní, dle mého názoru člověk vždycky raději komunikuje s dalším člověkem než s digitálním formulářem. Odpovědí od respondentů přišlo málo a většinou byly strohé a stručné.

Zvolil jsem osobní přístup a s každým dalším respondentem jsem komunikoval jako k sobě rovným. Členové umělecké komunity spolu komunikují většinou v anglickém jazyce. Pro maximalizaci šance, že respondent odpoví jsem ve zprávě zmínil fakt, že zásuvné moduly vytvořené v této bakalářské práci mohou zlepšit produktivitu právě jim. Na rozdíl od zkušenosti s formulářem se mi dostalo mnohem přívětivějšího přístupu.

## 2.4 Výsledky průzkumu

Ze 71 umělců, které jsem oslovil, jsem navázal písemnou korespondenci s 22 z nich.

### 2.4.1 Rozbor odpovědí – Pracovní postup

Myslel jsem si, že někteří umělci budou svůj pracovní postup brát jako součást jejich duševního vlastnictví a nebudou se chtít podělit, ale všichni respondenti, kteří na otázku odpověděli, byli otevření a svůj pracovní postup popsali.

Z odpovědí jsem zjistil, že zhruba polovina respondentů se specializuje spíše na práci ve 3D softwaru a druhá polovina respondentů naopak na práci ve 2D softwaru. Tato hranice se liší projekt od projektu, protože různí klienti mají různé požadavky. Pro rychlé skici nebo koncepty se práce se 3D nehodí, ale pro práci na portfoliu je ideální. Pracovní postup záleží na zadaném úkolu [12].

Jeden z respondentů také popsal zajímavý poznatek z jeho pracovního postupu: „*V konceptním umění je čas klíčový. Vytváření mnoho nápadů rychle je nezbytné v našem procesu, takže radši než strávit moc času ve 3D, jednoduše skočím do Photoshopu a překreslím chyby. Občas to může ušetřit hodiny, narozdíl od snahy opravit problém ve 3D.*“ [12]

### Pracovní postup Blender-Photoshop

Pro další analýzu bylo třeba sestavit největší průnik pracovních postupů členů cílové skupiny. 20 z 21 respondentů popsal svůj pracovní proces a z těchto odpovědí jsem sestavil postup, na který se zaměřím v praktické části. Postup lze vidět na obrázku č. 1.2. Tento postup samozřejmě neplatí pro všechny, digitální malba je z velké části o individuální práci, takže postupy se liší

umělec od umělce. V obrázku jsem se snažil zobecnit postup, aby byl validní pro co největší počet respondentů, kteří svůj postup specifikovali.

Pracovní postup je zobrazen na obrázku č. 1.2. Ve Photoshopu umělec načrtne skicu hrubé scény, dále se přesune do Blenderu, kde na základě náčrtů vytvoří 3D scénu [12]. Objektům přidá materiály, textury, osvětlení a různé ambientní efekty. Jakmile je scéna hotová, má umělec prostor zkoušet různé kamerové pozice a nastavení kompozice. Poté vyrenderuje finální obrázek společně s kompozičními vrstvami a manuálně je importuje do Photoshopu. Ve 2D pak umělec upravuje jednotlivé kompoziční vrstvy, dokresluje objekty a postavy, ladí detaily.

## 2.4.2 Rozbor odpovědí – Úskalí pracovního postupu

Odpovědi respondentů na druhou otázku mi nabídly podměty pro tvorbu zásuvných modulů.

Problém práce v Blenderu popsal jeden z respondentů: „3D prostředí dokáže být velmi technické, což může člověku zatemnit schopnost přemýšlet kreativně, protože tam je tolik věcí, na které je třeba myslet.“ [12]

Přesně tento problém se snažím řešit v této práci. Pokud se umělec musí plně soustředit na komplikované technické úkoly, zbude méně času na kreativitu. Pokud zásuvný modul pomůže uživateli automatizovat některé technické úkoly, nabídne mu tím myšlenkovou kapacitu pro kreativitu.

Konkrétní problémy postupu Blender-Photoshop, které sužují respondenty, popisují v kapitole Analýza.

## 2.4.3 Rozbor odpovědí – Používané zásuvné moduly

Z odpovědí na otázku č. 3 lze vyzorovat důležité informace o způsobilosti programů pro pracovní postup Blender-Photoshop. Z 16 respondentů, kteří popsal, jaké moduly používají 13 odpovědělo, že používá zásuvné moduly v aplikaci Blender, ale jen 2 odpověděli, že používají alespoň jeden zásuvný modul v aplikaci Photoshop. Výsledný nepoměr může ovlivňovat hned několik faktorů. Prvním faktorem je, že Blender Python API má propracovanější dokumentaci, obecně snazší proces tvorby zásuvných modulů, a proto je jejich tvorba pro vývojáře lákavější. Dalším důležitým faktorem je cena programu. Photoshop má velmi drahé měsíční poplatky. Aby člověk mohl používat UXP Developer Tool pro vývoj zásuvných modulů (použití této aplikace při tvorbě zásuvných modulů není povinné, ale je pohodlné), musí vlastnit licenci na Adobe Creative Cloud, a proto vývoj zásuvných modulů pro Photoshop je dražší než pro Blender.

V kontextu mého průzkumu, nejčastěji používaným zásuvným modulem pro Blender je Boxcutter od vydavatele TeamC. Zásuvný modul Boxcutter lze zakoupit za dvacet dolarů na serveru BlenderMarket. Boxcutter nabízí nové nástroje na úpravu geometrie objektů. Práce s ním je rychlá a intuitivní, takže Boxcutter může používat i uživatel bez hlubších technických znalostí.

Dalším často zmíněným zásuvným modulem pro Blender je Sketchfab. Přidává uživateli možnost v prostředí Blenderu procházet rozsáhlou Sketchfab databází 3D modelů. Platforma Sketchfab slouží primárně jako online tržiště, ale v roce 2021 překonala hranici 600 000 bezplatných modelů [13].

Z odpovědí jsem se dozvěděl, že mezi často používané zásuvné moduly pro Photoshop patří Chameleon Adaptive Palette. Má spoustu užitečných funkcí, rozšiřující možnosti výběru barev. S jednoduchým UI vytváří gradienty již vybrané barvy s dalšími barvami. Tento proces simuluje způsob, jakým se míchají barvy v tradiční malbě, který je pro nás daleko přirozenější než vybrat novou barvu z duhové škály.



## 2.4.4 Rozbor odpovědí – Kompoziční vrstvy a OpenEXR

Z 18 respondentů popisující použité kompoziční vrstvy, 15 odpovědělo, že využívají více kompozičních vrstev, 2 odpověděli, že používají pouze clown pass pro rychlý výběr oblasti. Z 18 odpovědí, pouze 4 respondenti odpověděli, že používají formát OpenEXR. Zbylých 14 respondentů používají různé kompoziční vrstvy a ukládá výstupy manuálně. V dalších odstavcích reaguji na konkrétní odpovědi týkající se kompozičních vrstev a OpenEXR formátu.

*„Ano, vždycky používám více kompozičních vrstev, jsou průlomové ve fázi překreslování, dokáže to výsledné dílo dostat na další úroveň. Pravdou ale je, že OpenEXR nedokáže v Blenderu zprovoznit, dříve jsem ho často používal s Octane, Redshift, Vray nebo Corona (Render enginy, které Blender nepodporuje). Ale ne v Blenderu, v něm musím ukládat každou kompoziční vrstvu zvlášť!“ [12]*

Respondent poukazuje na fakt, že renderování OpenEXR formátu v Blenderu není intuitivní. Je třeba k tomu vizuálním programováním vytvořit kompoziční uzly se správným nastavením a zapojením. Jedná se jistě o další potenciální místo pro optimalizaci.

*„V současnosti vyrenderuji vícevrstvý EXR a pokaždé kopíruji všechny vrstvy do Photoshop souboru! Přemýšlel jsem, že most mezi těmito dvěma aplikacemi by byl skvělý, hodně by zrychlil můj pracovní proces, dovolil by mi přepínat tam a zpět mezi renderováním a 3D a současně by ponechal úpravy, které jsem udělal ve Photoshopu!“ [12]*

Respondent popsal jedno z největších úskalí tohoto pracovního postupu. Prvním místem pro potenciální optimalizaci je přesun dat mezi aplikacemi. V ideálním případě by stačil jeden klik a nově vyrenderované výstupy by se importovaly do Photoshopu.

*„V minulosti jsem experimentoval s cryptomatte (způsob tvorby masek), ale usoudil jsem, že Blender je s celým procesem otravný. Hrátky s uzly v kompozičním oknu jsou moc problematické a berou moc času. Přál bych si, aby byla cesta, jak ukládat cryptomatte masky závislé na kolekcích.“ [12]*

Respondent míní ukládat sloučené masky objektů ve stejných kolekcích. Pixelové masky, které cryptomatte data vytváří, mohou být použity místo clown passu a to by šetřilo čas při práci ve Photoshopu.

*„Jenda z věcí, které mě štve v Blenderu je generování „objekt passu“ (respondent míní clown pass), víte ten render, na kterém má každý objekt jinou jednodílnou barvu. Nejhorší na tom je, že Blender nepoužívá dost barev a několik objektů mají přesně tu samou barvu.“ [12]*

Nechtěný důsledek, který respondent zmiňuje tkví v následném použití clown passu v Photoshopu. Označení nástrojem „magická hůlka“ vybírá automaticky na základě barevné informace a proto, když jsou v clown passu dva objekty se stejnou barvou vedle sebe, nelze označit jen jeden z nich. Blender pro renderování clown passu nemá žádné rozšiřující nastavení.

## 2.4.5 Rozbor odpovědí – Opětné renderování

*„Je problémové, když klient žádá malé změny a já musím přerenderovat celou scénu a ano, je to velmi časově náročné. Protože potom musím znovu přidat kompoziční vrstvy se všemi změnami, které jsem udělal, jako blending mód, úpravu barev a masky. Řekl bych, že tohle je nejvíce otravný problém.“ [12]*

Respondent popisuje klasický problém pracovního procesu Blender-Photoshop. V tomto procesu nelze aplikace používat souběžně. Umělec vytváří scénu, vyrenderuje ji a pak může s výslednými výstupy začít s úpravami ve Photoshopu. Umělci často pracují s rendery destruktivně. To znamená, že upravují přímo vyrenderované vrstvy a když je třeba změnit scénu ve 3D a vyrenderovat výstupy znovu, musí všechny změny nad těmito vrstvami udělat znovu. Nejlepším řešením je samozřejmě pracovat s výstupy 3D programu nedestruktivně, všechny změny dělat na nové vrstvy a nechat výstupy z 3D programu nezměněné. Bohužel, ne v každém případě je to možné.

*„Občas renderuji obrázek s objekty v popředí, o kterých vím, že je nejspíš budu ve fázi přemalování přesouvat pro lepší kompozici a zároveň renderuji bez těchto objektů, abych měl vykreslené pozadí za nimi. Bylo by skvělé, kdyby Blender uměl renderovat některé objekty na separátní vrstvu najednou. Zároveň by bylo skvělé, kdybych si mohl zvolit kolekce s objekty, které by měly být na odlišné vrstvě.“ [12]*

Respondent popisuje problém zapečení 3D objektů do 2D renderu. První část, popisující nutnost dvojího renderování má potenciál na automatizaci s pomocí zásuvného modulu, ale renderování objektů na separátní vrstvy má svá úskalí, která popisují v sekci Problémy pracovního postupu.



## Kapitola 3

# Analýza

*V následující kapitole Zkoumám různé způsoby tvorby zásuvných modulů pro aplikace Blender a Photoshop. Dále pomocí výsledků průzkumu vybírám vhodné kroky pracovních postupů k optimalizaci zásuvnými moduly a specifikuji jejich funkční a nefunkční požadavky.*

### 3.1 Programování s knihovnou Blender Python API

Pro lepší porozumění tvorby zásuvných modulů pro Blender je třeba si vysvětlit několik klíčových konceptů Blender API. Zásuvné moduly (Python skripty) lze integrovat do aplikace Blender více způsoby [14]:

- Jako definici nového render engineu
- Jako definici operátorů
- Jako definici nového menu, záhlaví nebo panelu
- Jako přidání nových prvků do již existujících menu, záhlaví nebo panelů

### Metadata

Na začátku modulu je třeba definovat informace o modulu ve formě vyplnění python dictionary `bl_info`. Tento krok je povinný pro každý zásuvný modul určený aplikaci Blender. Mezi povinné atributy patří „name“, jméno zásuvného modulu, atribut „blender“, který specifikuje minimální podporovanou verzi Blenderu a atribut „category“, který určuje kontext zásuvného modulu. Ostatní atributy jsou nepovinné a slouží k bližší specifikaci. Informací „blender“ specifikují, pro jakou verzi Blenderu modul vyvíjím, ale stejná verze není nutnou podmínkou. Modul je možné používat i na starších verzích, jen není garantováno, že bude fungovat správně. Informace „location“ specifikuje, kde v aplikaci modul bude a zároveň i jaký bude jeho kontext. Nakonec informace „support“ specifikuje, kdo má na starosti údržbu modulu.

### Přístup k datům

Přístup k datům přes Python API funguje stejně jako nativní uživatelské rozhraní; Tohle implikuje, že jakékoliv nastavení, které může být změněno pomocí tlačítka uživatelského rozhraní, může být změněno přes Python API.

Data nahraného projektu jsou přístupná přes modul `bpy.data`. Tento modul nám zpřístupňuje objekty, scény, materiály a mnoho dalších důležitých dat přímo ve skriptu a umožňuje upravovat jejich atributy. Datům lze skrze `bpy.data` přidávat vlastní proměnné atributy.

```
# Data can be accessed via name or index
bpy.data.objects['Cube']
# Creating new attribute
bpy.data.objects[0]["MyNewProperty"] = 23
```

■ **Výpis kódu 3.1** Výpis kódu č. 1 (Python)

## Kontext

Dalším důležitým konceptem je `bpy.context`. Jedná se o strukturu, která v sobě drží aktuální hodnoty kontextu. Kontext i jeho atributy se mění na základě okna uživatelského rozhraní, ve kterém se uživatel nachází nebo se kterým interaguje.

## Jednoduchý příklad

Pro demonstraci jsem vytvořil velmi jednoduchý modul „Hello World“, který pouze pozdraví uživatele v grafickém rozhraní. Modul lze najít v elektronické příloze na cestě `/API_tests/blender_hello_world.py`. Po instalaci jej lze najít v aplikaci v okně View 3D v kategorii Tools.

## Instalace modulu pro Blender

Instalace zásuvných modulů s Blender Python API je velmi snadná. Otevřu dialog v aplikaci na cestě `Edit > Preferences... > Add-ons > Install...`. V tomto dialogu vyberu Python skript z příloženého elektronického média. Jakmile se zásuvný modul nahraje, objeví se v nabídce. Povolím jej zaškrtnutím políčka. Pokud se políčko zaškrtně a nevyskočí žádné chybové hlášky, modul je správně nainstalován a lze přejít rovnou k jeho použití bez restartování aplikace. V opačném případě nastala chyba a je třeba modul upravit a instalaci zopakovat.

### 3.2 Výběr způsobu tvorby modulu pro Photoshop

Jak jsem již popsal v sekci Zásuvné moduly pro grafické aplikace, Photoshop nabízí dvě knihovny, se kterými lze tvořit zásuvné moduly. Dal jsem si za úkol prozkoumat každou variantu a vytvořit si jednoduchý „Hello world“ zásuvný modul každým dostupným způsobem.

## Photoshop SDK

Při tvorbě jednoduchého řešení s knihovnou Photoshop SDK v jazyce C++ jsem postupoval podle návodu od Siew Yi Liang [15]. Vytváření modulu s touto knihovnou není jednoduché ani intuitivní. Společně se špatnou dokumentací tvoří uživatelsky nepříjemný proces. Použití knihovny Photoshop SDK vyžaduje použití velmi složitého kompilovacího procesu [15], který má na výstupu DLL knihovnu s možnými formáty `.8li` nebo `.8bf`. Podařilo se mi vytvořit, zkompileovat a nahrát modul, který nic nedělá. Modul lze najít v příloze `/API_tests/SDK_hello_world`. Tento způsob tvorby modulu sice nabízí možnost importování, které potřebuji, ale proces tvorby jsem uznal za nevhodný.

Pro instalaci je třeba přidat soubor z elektronické přílohy na adrese `/API_tests/SDK_hello_world/PhotoshopPlugin.8bf` do instalačního adresáře Photoshopu do

podadresáře „Plug-ins“ a zásuvný modul bude přístupný v aplikaci v záložce „Filtry“ pod jménem „PhotoshopPlugin“.

## Photoshop UXP API

Tvorba s pomocí knihovny Photoshop UXP API je o něco jednodušší. Není třeba žádné kompilace a existují příručky, přímo od společnosti Adobe (<https://github.com/Adobe-CEP/Getting-Started-guides>), které pomáhají začátečníkům při tvorbě modulu. Navíc Adobe nabízí aplikaci Adobe UXP Developer Tool, která výrazně pomáhá při ladění chyb. S pomocí příkladových řešení jsem vytvořil zásuvný modul s uživatelským rozhraním, který lze najít v elektronické příloze /API\_tests/UXP\_hello\_world. Zásuvný modul vypisuje „Hello World“ a zároveň má dvě tlačítka „Click“ a „Clear“. Modul vypisuje kolikrát uživatel stiskl tlačítko „Click“ a tlačítko „Clear“ počítadlo resetuje. Tato knihovna je pro mé účely použitelná, ale mé řešení nevyžaduje stále grafické rozhraní.

Protože zásuvný modul není vydaný, pro instalaci je třeba si stáhnout Adobe UXP Developer Tool. V aplikaci stisknout „Přidat Zásuvný modul“ a vybrat soubor z elektronické přílohy /API\_tests/UXP\_hello\_world/manifest.json. Jakmile se modul nahraje, je třeba spustit aplikaci Photoshop a pak v Developer Tool u modulu v nastavení stisknout „Load“. Modul se automaticky spustí v otevřeném Photoshopu. Minimální verze je nastavena na 22.0.

### 3.3 Příklady současných řešení

Zásuvných modulů pro grafické aplikace existuje velké množství. Pro účely této práce je třeba zúžit výběr již existujících zásuvných modulů.

Důležitým indikátorem je užití v průmyslu. Z informačních stránek zásuvných modulů tuto informaci nelze zjistit, a proto jsem toto téma zařadil do průzkumu (Otázka č. 3). Z výsledků průzkumu jsem zjistil, že žádný z respondentů nepoužívá zásuvné moduly ani skripty pro export a import.

#### 3.3.1 Photoshop

Společnost Adobe Inc. podporuje tvorbu zásuvných modulů a zároveň jejich distribuci. Na platformě Creative Cloud spravující veškeré aplikace od Adobe Inc. lze procházet nabídku zásuvných modulů přímo od Adobe Inc nebo od externích vydavatelů. Na této platformě lze zásuvný modul přímo nainstalovat, a dokonce jej hodnotit.

Actions je nová a velmi užitečná funkcionality Photoshopu. Uživatel si může nahrát konkrétní pracovní procesy a následně je spouštět. S touto funkcionalitou lze vytvářet sady instrukcí, které se budou provádět automaticky. V době před vydáním této funkcionality, byla automatizace možná pouze se skriptem nebo zásuvným modulem, tímto krokem však Adobe Inc. výrazně snížilo potřebu tvořit nové zásuvné moduly.

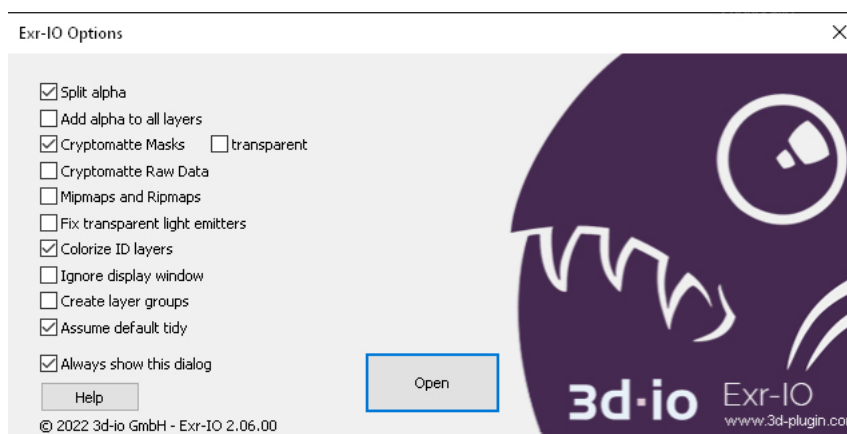
#### Import Folder As Layers

Skripty od Travora Morrise jsou volně dostupné na webové stránce (<https://morris-photographics.com/photoshop/scripts/index.html>). Většina z nich jsou určeny pro úpravy fotografií, ale skript `import_folder_as_layers.jsx` je zajímavý i v případě pracovního procesu Blender-Photoshop. Tento skript importuje obsah zvolené složky do nového Photoshop dokumentu. Podporuje import formátů PNG, GIF, JPEG, BMP, TIF a dokonce i formát Photoshop dokumentu PSD (Obsah dokumentu sloučí do jedné vrstvy a tu duplikuje do nového dokumentu). Po importu všech obrázků do nového dokumentu se upraví rozměry dokumentu, aby se do něj všechny vrstvy vešly. Toto řešení má nechtěné vedlejší účinky. Výsledkem

je, že importované soubory jsou v novém dokumentu, tudíž uživatel je stejně musí zkopírovat do svého aktivního dokumentu. Nový dokument má větší rozměry než importované obrázky, to vytváří prázdné místo. Všechny vrstvy jsou zarovnané do pravého dolního rohu. Tento skript nemá ve své formě přínosné využití pro pracovní proces Blender-Photoshop, ale lze se inspirovat tímto řešením pro novou implementaci.

## EXR-IO

EXR-IO patří do sady zásuvných modulů pro export a import formátu OpenEXR. Konkrétně zásuvný modul pro Photoshop slouží k importu formátu OpenEXR do Photoshop dokumentu. Manuál tvrdí, že po instalaci stačí otevřít OpenEXR formát a objeví se dialogové okno s několika možnostmi nastavení importu, jenže některé novější verze Photoshopu toto chování nepodporují a upřednostňují nativní importní dialog. Po aktualizaci na verzi Photoshopu 24.0 se mi podařilo dialogové okno zobrazit. Zásuvný modul umožňuje různé nastavení importu jako třeba přidání alfa kanálu do každé vrstvy nebo nastavení bitové hloubky barev (viz. obrázek č. 3.1).



■ **Obrázek 3.1** Nastavení modulu EXR-IO při importu

### 3.3.2 Blender

Pro vývoj zásuvných modulů v Blenderu byla vytvořena knihovna Blender Python API. Vydavatelé Blenderu nabízí rozsáhlou sadu zásuvných modulů, které si lze aktivovat přímo v aplikaci. Pro všechny ostatní zásuvné moduly je třeba využít jeden z externích serverů. Jedním z nej-používanějších serverů je BlenderMarket z produkce CGCookie, který funguje jako online tržiště pro uživatele Blenderu.

#### Exr Auto-Pass Saver

Tento skript patří mezi sadu zásuvných modulů od vydavatele Exr-IO dostupný na adrese (<https://www.exr-io.com/automate-blender-exr-elements/>). Je napsaný v Pythonu s využitím Blender Python API a automaticky generuje kompoziční uzly potřebné k exportu kompozičních vrstev. Zásuvný modul nijak neupravuje kompoziční vrstvy, pouze je společně zapouzdří a exportuje ve formátu Multilayer OpenEXR. Celá sada Exr-IO je volně k dispozici na internetu a spadá pod 3-bodovou BSD licence. To mi dává příležitost inspirovat se řešením v implementaci mého zásuvného modulu.

Výhodou použití formátu OpenEXR je, že jej lze importovat do Photoshopu a tím otevřít nový dokument se všemi kompozičními vrstvami najednou. Zásuvný modul má však několik nevýhod.

První je, že kompoziční vrstvy nejsou odšuměné. druhou nevýhodou jsou pak nepřehledná jména, která se vrstvám automaticky přiřazují a často s účelem kompoziční vrstvy nemají nic společného.

## Build Export Node

V průběhu psaní této práce jsem narazil na zásuvný modul Build Export Node od vydavatele Bookyakuno, který automaticky vytváří uzly v Kompozitoru pro export kompozičních vrstev. Tento modul nabízí některé funkcionality mého implementačního řešení. Dá se koupit za patnáct dolarů na serveru BlenderMarket (<https://blendermarket.com/products/build-export-node?ref=2>).

### 3.4 Problémy pracovního postupu Blender-Photoshop

Proces exportu kompozičních vrstev z Blenderu je často opakovaná činnost, která bohužel není bez znalosti vizuálního programování s kompozičními uzly snadná [12].

Z výsledků průzkumu lze odvodit několik problémů, které lze řešit zásuvnými moduly. V následující sekci popisují postupy, ve kterých se nachází problém a jaké jsou důsledky těchto problémů.

## Maskování objektů

Z odpovědí je zřejmé, že i když je clown pass velmi používaný, není ideálním řešením. Ve Photoshopu vytváří nutnost označit objekt magickou hůlkou při každé změně. Navíc ve chvíli, kdy uživatel upraví pozici objektu ve Photoshopu, celý clown pass ztácí validitu. Další nevýhodou renderování clown passu s náhodnou barvou je, že občas vytváří situace, kdy dva objekty vedle sebe mají stejnou barvu, a to znemožní správné označení objektu. Žádný z těchto problémů by neexistoval, pokud by uživatel měl některé objekty na separátních vrstvách [12].

## Export kompozičních vrstev z Blenderu

15 respondentů odpovědělo, že renderují ve svém pracovním procesu více než jednu kompoziční vrstvu. Při použití alternativního způsobu renderování bez použití kompozičních uzlů se kompoziční vrstvy vyrenderují automaticky, ale uživatel je musí uložit manuálně. Při manuálním ukládání musí u každé kompoziční vrstvy zvolit jméno a místo uložení. 4 z 18 respondentů odpověděli, že pro přesun dat z Blenderu do Photoshopu používají OpenEXR formát. Ten umožňuje uložit všechny kompoziční vrstvy do jednoho souboru a eliminuje ztracený čas manuálním ukládáním. OpenEXR se zdá jako správné řešení, ale práce s ním má nevýhodu. Uživatel si nemůže zkontrolovat výstupy před importováním do Photoshopu, protože OpenEXR nemá nativní prohlížeč v systémech Windows, macOS ani Linux. Zároveň není možné upravovat složení OpenEXR formátu před importem do Photoshopu. Tyto skutečnosti mě přesvědčili, že při automatizaci exportu obrázku je použití jednotlivých výstupů v klasických obrázkových formátech užitečnější.

## Import kompozičních vrstev do Photoshopu

Při využití OpenEXR formátu se importují všechny kompoziční vrstvy naráz, ale i tak musí uživatel z nového Photoshop dokumentu s importovanými vrstvami zkopírovat všechny vrstvy po jedné do aktuálního souboru. Tento proces je obzvlášť zdlouhavý při opětovném renderování, kdy je třeba staré vrstvy vyměnit za nové. Nový zásuvný modul pro Photoshop by mohl automaticky importovat všechny obrázky ze složky. Dalším nápadem pro optimalizaci je vytvořit nástroj pro nahrazení vrstvy importovanou vrstvou [12].

V současné nejnovější verzi Photoshopu (24.0) má uživatel dvě možnosti, jak importovat více obrázků najednou. První je metoda drag & drop. Obrázky se po jednom importují jako chytré objekty. Chytré objekty chrání rozměry a kvalitu importovaného obrázku, ale to je v našem případě nepotřebné, protože všechny kompoziční vrstvy mají stejné rozměry i rozlišení. Výsledkem je, že uživatel musí manuálně rastrovat každou importovanou vrstvu. Druhým způsobem je klasický postup přes funkci „Otevřít“. Po otevření dialogového okna vybere všechny obrázky, které chce importovat a potvrdí svůj výběr. Problém tohoto postupu je, že jednotlivé obrázky importuje každý do nového Photoshop dokumentu, takže uživatel musí z každého dokumentu duplikovat ručně obrázek na svůj aktivní dokument.

Pro zrychlení importu více kompozičních vrstev do Photoshopu je třeba vytvořit novou funkcionalitu, která automaticky importuje všechny obrázky jako vrstvy, bez dalších nutných operací. Pro tento účel není nutné vytvářet zásuvný modul se stálým uživatelským rozhraním, funkcionalitu lze integrovat jako Photoshop Script v jazyce JavaScript. Aby skript ušetřil další potenciační kroky pracovního postupu, lze dát uživateli možnost nahrazení vrstev se stejným jménem. Náhrada kompozičních vrstev je nutným krokem při fázi opětovného renderování.

## Zapečení 3D scény v renderu

Problém popsal respondent v odpovědi na otázku opětovného renderování [12]. Nápadem, který by mohl omezit nutnost opětovně renderovat výstupy z 3D programu, je, renderovat některé objekty na separátní vrstvy a nechat vykreslit místa, která blokují. Tímto by objekty bylo možno přesouvat ve 2D softwaru a nebylo by nutné kvůli každé změně opětovně renderovat všechny výstupy.

Tento nápad má však překážky. Pokud chci vyrenderovat objekt na separátní vrstvu, vytratí se jeho interakce se scénou. U menších změn by toto nemuselo být problémem, ale u větších změn je opětovný render nutný pro zachování integrity scény. Tento problém lze popsat na jednoduchém případu.

Mám kostku na ploše a světlo nastavené tak, aby kostka vrhala na plochu stín. Pokud vyrenderuji kostku na separátní vrstvě, render nebude obsahovat stín, který vrhá na plochu, protože na této vrstvě žádná plocha není. Žádný stín kostky nebude ani na vrstvě plochy, protože kostka se renderuje na jiné vrstvě. Pokud tedy tyto dva rendery importuji do Photoshopu a dám na sebe, kostka nebude světelně interagovat s okolními objekty. Do renderů se nepromítne ani vrhající stín kostky, ani odražené světlo od povrchu ostatních objektů.

I když tento způsob má svoje vady, existují situace, ve kterých jsou tyto vady zanedbatelné. Zároveň v jedné z odpovědí respondent potvrdil, že řešit chybějící pozadí objektu ve Photoshopu je problém, ale překreslit pár stínů a světelných interakcí nikoliv. Z toho jsem usoudil, že má smysl řešení implementovat.

## Odšumění kompozičních vrstev

Šum na kompozičních vrstvách vzniká jako vedlejší efekt metody Monte Carlo Path Tracing, kterou používá render engine Cycles. Blender podporuje použití knihovny OpenImageDenoise, které eliminují tyto vedlejší efekty, a zmenšují tak počet nutných vzorků při renderování [16]. Pokud má uživatel v Blenderu povolený „Denoise“ a renderuje různé kompoziční vrstvy, procedura odstranění šumu se automaticky provádí pouze na kombinované vrstvě. Všechny ostatní kompoziční vrstvy nepodléhají odšumění a tím pádem jsou často odlišné od finálního obrázku. Aby uživatel překonal tento problém, musí v Kompozitoru složitě vytvářet uzly na odstranění šumu pro každý výstup zvlášť. Tento úkol je jistě možné automatizovat zásuvným modulem.

## 3.5 Funkční a nefunkční požadavky

### F1.1: Automatické generování kompozičních uzlů pro export

Bude možné stiskem tlačítka vygenerovat potřebné kompoziční uzly pro renderování a export kompozičních vrstev. Tento proces nebude požadovat žádnou interakci uživatele s kompozičními uzly a bude spustitelný jedním tlačítkem. Výběr kompozičních vrstev uživatel provede před použitím zásuvného modulu.

### F1.2: Automatické renderování a export

Bude možné po vygenerování kompozičních uzlů automaticky spustit renderování kompozičních vrstev a po úspěšném vyrenderování výstupy automaticky exportovat na lokální disk. Nastavení renderování uživatel provede před použitím zásuvného modulu.

### F1.3: Renderování clown passu

Bude možné k automatickému renderování přidat clown pass, který se renderuje s pomocí Workbench render enginu. Na stisknutí jednoho tlačítka lze nastavit renderovací vlastnosti Workbench render enginu a zároveň spustit renderovací proces. V Blenderu není možné paralelně renderovat ze dvou rozdílných render enginů, takže renderování a export proběhne sekvenčně.

### F1.4: Renderování pozadí objektů

Bude možné k automatickému renderování přidat kombinovanou vrstvu dalších dostupných render vrstev. Výstupní soubory převezmou jména render vrstev. Veškerá nastavení dalších render vrstev provede uživatel před použitím zásuvného modulu.

### F1.5: Export výstupů do složky

Bude možné specifikovat složku na lokálním disku, do které se budou výstupy z renderování automaticky ukládat. Pokud uživatel zvolí složku, která na lokálním disku neexistuje, proces novou složku vytvoří.

### F1.6: Maskování kolekcí

Bude možné automaticky aplikovat společné pixelové masky pro objekty ve stejné kolekci. Výsledný výstup s aplikovanou maskou vždy převezme jméno kolekce a exportuje se společně s kompozičními vrstvami. Organizací objektů ve scéně do správných kolekcí provede uživatel před použitím zásuvného modulu. Kolekce, které obsahují pouze světla a kamery budou přeskočeny.

### F1.7: Intuitivní pojmenování výstupních souborů

Bude možné zvolit společný prefix všech výstupních souborů. Jména konkrétních kompozičních vrstev budou vždy odpovídat jejich pojmenování v Blenderu.

### F1.8: Kontrola výstup

Bude možné po vyrenderování automaticky otevřít složku na lokálním disku s exportovanými výstupy. To umožní uživateli zkontrolovat proces před importem do Photoshopu.

### F1.9: Rychlý úklid kompositoru

Bude možné na jedno zmáčknutí tlačítka smazat všechny kompoziční uzly v kompositoru. Zásuvný modul nerozlišuje mezi vygenerovanými uzly a uzly vytvořené uživatelem.

### F1.10: Nastavení výstupních formátů

Bude možné nastavit výstupní formát každého exportovaného souboru. Výchozím nastavením bude formát PNG. Podporovanými formáty budou JPG, PNG a Multilayer OpenEXR.

### F1.11: Odšumění kompozičních vrstev

Bude možné automaticky aplikovat odšumění na všechny kompoziční vrstvy. Výstupy z cryo-matte maskování ani finální kombinovaná vrstva odšumění nepotřebují, jsou přeskočeny. Odstranění šumu funguje pouze s výchozí metodou OpenImageDenoise a zvoleným nastavením „Albedo and Normal passes“ v podrobnostech render engine Cycles.

### NF1.1: Podporovaná verze Blenderu

Zásuvný modul nebude podporován na verzích Blenderu starší než verze 2.93.

### F2.1: Automatický import všech obrázků ze složky

Bude možné zvolit složku na lokálním disku a z této složky automaticky importovat všechny obrázkové soubory. Import bude podporovat formáty PNG, JPG, BMP, TIF a GIF. Obrázky se importují jako vrstvy aktuálního dokumentu. Vrstvy nepodlehnu žádné transformační úpravě a převezmou jméno importovaných souborů.

### F2.2: Volání bez otevřeného dokumentu

Bude možné spustit skript bez otevřeného dokumentu. V takovém případě skript vytvoří nový dokument a po dokončení importu spustí dialog uložení. Nový dokument bude mít rozměry prvního importovaného obrázku.

### F2.3: Náhrada existujících vrstev

Bude možné při importu automaticky nahradit již existující vrstvy se stejným jménem. Po vybrání exportní složky se zobrazí dialog, ve kterém uživatel zvolí, zda chce vrstvy nahradit nebo ne.

### NF2.1: Podporovaná verze Photoshopu

Zásuvný modul nebude podporován na verzích Photoshopu starších než verze 8 (První verze CS z roku 2003).



*V následující kapitole navrhuji prototypy zásuvných modulů pro Blender a Photoshop na základě funkčních a nefunkčních požadavků, které jsem zvolil v kapitole Syntéza.*

Zásuvné moduly jsem vyvíjel primárně pro systém Windows, avšak mohou být použity i na systémech macOS a Linux bez větších problémů.

### 4.1 Zásuvný modul pro Blender

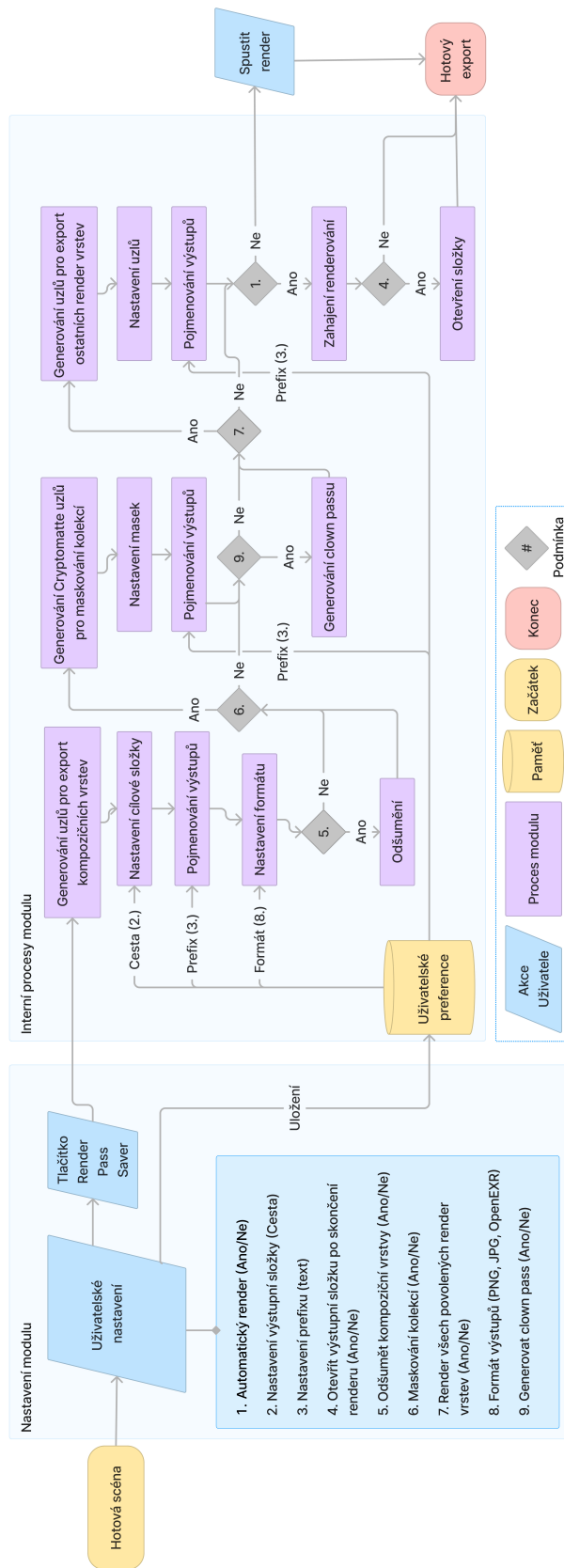
Při vytváření zásuvného modulu pro Blender jsem vyplnil informace o modulu (tzv. meta info). Tyto informace slouží k identifikaci modulu a k definici jeho požadavků.

- name: Auto Render Pass Saver
- description: Exports render passes and cryptomatte masks by collection in given location.
- blender: 3.1.0
- category: Compositioning
- version: 1.0
- location: Compozitor Tab - Sidebar - Auto Render Pass Saver
- support: COMMUNITY

Na obrázku č. 4.1 jsem popsal průběh procesů zásuvného modulu a jeho interakci s uživatelem. Počáteční stav jsem nazval „Hotová scéna“. V tuto chvíli uživatel zobrazil uživatelské rozhraní modulu v okně Kompozitoru. Uživatel musí před spuštěním modulu specifikovat, které kompoziční vrstvy chce vyrenderovat, zásuvný modul toto nastavení považuje za hotové.

Následuje fáze nastavení modulu. Jako první uživatel vybere cestu ke složce, do které se exportují všechny výstupní soubory (F1.5). Výchozí nastavení bude obsahovat cestu `C:\\tmp\\`. Řešení je navržené, aby se výstupy automaticky ukládaly a uživatel tak nemusel ukládat každou kompoziční vrstvu zvlášť.

Dalším prvkem nastavení je prefixu výstupních souborů (F1.7). Pro lepší organizaci uživatel specifikuje prefix (krátké slovo nebo zkratka). Tento prefix se při generování automaticky připojí ke jménu každého výstupního souboru. Tento prvek jsem přidal, aby bylo možné rozeznat jednotlivé várky kompozičních vrstev při opětovném renderování, pokud by se uživatel rozhodl je všechny ukládat na stejné místo.



■ **Obrázek 4.1** Diagram aktivit zásuvného modulu pro Blender

Dále si uživatel rozmyslí, které funkcionality zásuvného modulu chce povolit. Svou volbu bude mít možnost zadat formou zaškrtnutých políček. Zvolené možnosti se uloží do paměti modulu a zmáčknutím tlačítka „Render Pass Saver“ se automaticky spustí povolené procesy.

## Spuštění renderování

Funkcionalita pod názvem „Auto-render“ (F1.2) je první optimalizací. Pokud uživatel povolí tuto možnost, po vygenerování kompozičních uzlů se automaticky spustí renderovací proces.

## Otevření exportní složky

Další funkcionalitou je „Open Export Folder“ (F1.8). Pokud je povoleno, automaticky, po dokončení renderovacího procesu, otevře specifikovanou složku s výstupy. Uživatel bude moci po vyrenderování zkontrolovat, jestli jsou výstupy podle představ. Kontrola před importem do Photoshopu je možná jen pouze, pokud jsou výstupy v čitelných formátech. Výstup ve formátu OpenEXR nemá Windows ani macOS nativní prohlížeč, a proto není vhodným formátem.

## Maskování podle kolekce

Další funkcionalitou je „Mask out by collection“ (F1.6). Pokud je funkcionalita povolena uživatelem, proces generování kompozičních uzlů navíc vygeneruje uzly potřebné při tvorbě společných pixelových masek s pomocí cryptomatte dat scény. Každé kolekci v aktuální scéně se vytvoří a aplikuje společná maska všech jejích objektů. Aby funkcionalita správně fungovala, je třeba mít povolený CryptObjekt, jako součást nastavení kompozičních vrstev. Zároveň uživatel musí vytvořit specifické kolekce a naplnit je objekty, pro které chce aplikovat společnou masku. Pokud uživatel nechce některé objekty zahrnout do renderu, musí je schovat nebo přesunout do kořenové kolekce, pro kterou se proces neprovádí. Vnořené kolekce se budou generovat samostatně a jejich obsah se nebude renderovat do masky nadřazené kolekce. Výsledné masky budou pojmenovány podle kolekce, kterou maskují.

## Odšumění všech kompozičních vrstev

Další funkcionalitou je „Denoise all passes“ (F1.11). Povolení této možnosti při generaci kompozičních uzlů automaticky aplikuje odšumění na všechny kompoziční vrstvy.

## Renderování ostatních render vrstev

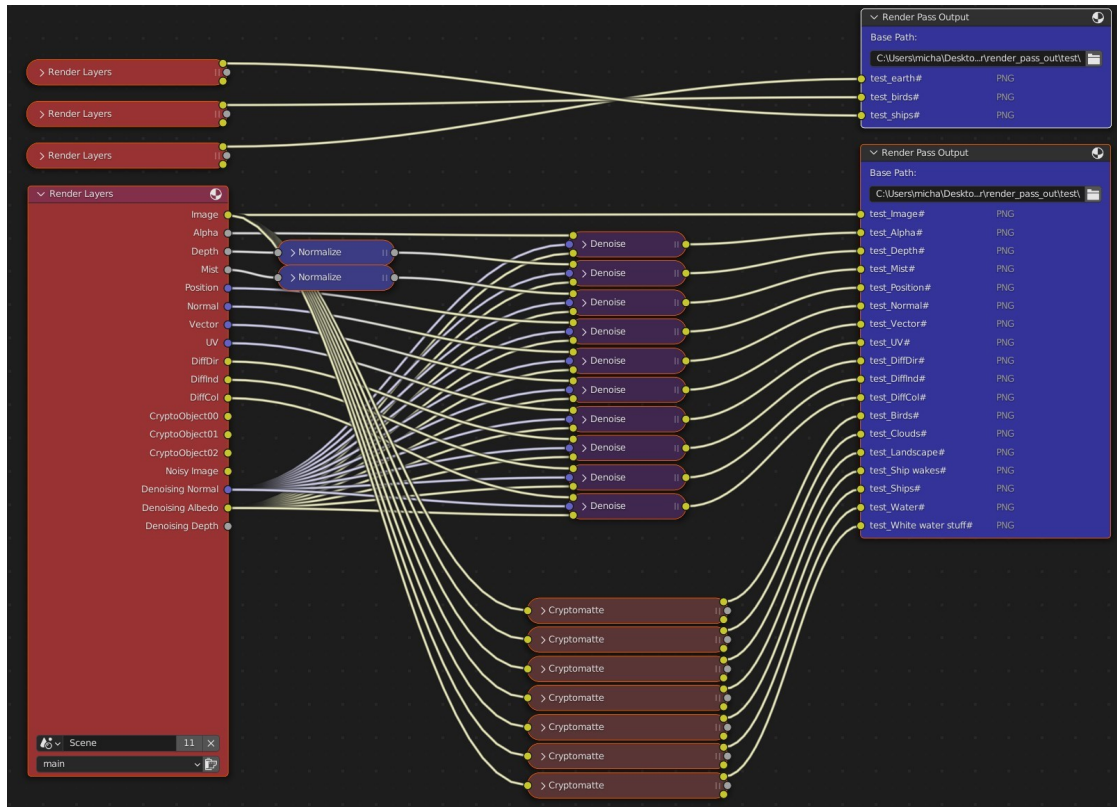
Poslední funkcionalitou je „Render combined from all layers“ (F1.4). Povoláním této možnosti lze k exportům přidat kombinované vrstvy všech ostatních render vrstev s povoleným renderováním. Uživatel může v ostatních vrstvách skrýt objekty v popředí, aby se na vrstvě vyrenderovaly části objektů, které jsou na hlavní vrstvě zakryté. To společně s cryptomatte maskami umožňuje manipulovat nedestruktivně s pozicí objektů ve Photoshopu. Pokud uživatel bude chtít tuto možnost využít, musí před generací kompozičních uzlů povolit renderování ostatních požadovaných render vrstev a nastavit jejich obsah.

## Rychlý úklid Kompozitoru

Zásuvný modul bude nabízet tlačítko „Clean All Nodes“ (F1.9). Tlačítko smaže všechny kompoziční uzly ve scéně. Nelze na něj kliknout, pokud uživatel nemá povolené renderování s použitím

kompozičních uzlů (Use nodes). Jedná se o doplňkovou funkcionalitu, takže v diagramu na obrázku č. 4.1 není popsána.

## Generace kompozičních uzlů



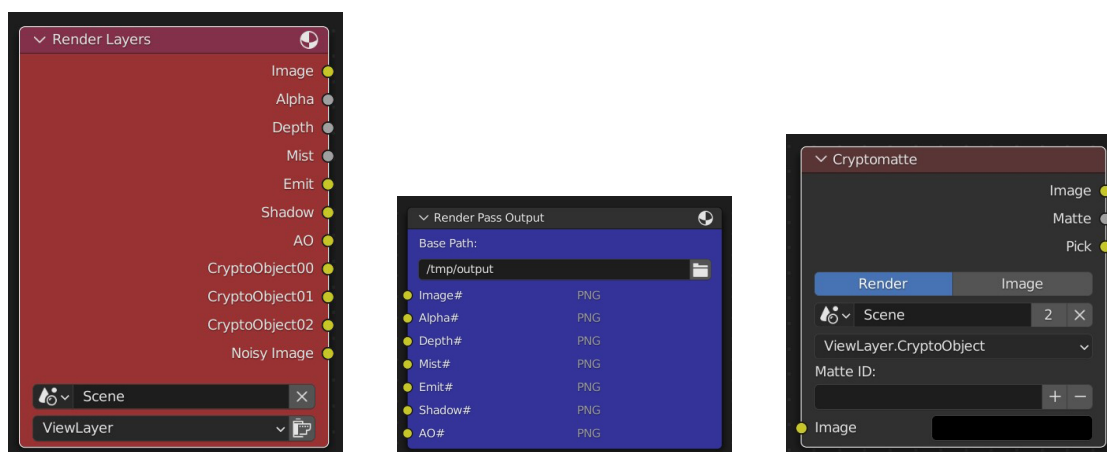
**Obrázek 4.2** Návrh zapojení kompozičních uzlů v Blenderu

Stisk tlačítka „Render Pass Saver“ automaticky spustí interní procesy zásuvného modulu, které vytvoří nutné kompoziční uzly pro export a dále i uzly potřebné k provedení povolených procesů specifikovaných uživatelem. Tyto uzly zásuvný modul přesune na pozice v Kompozitoru tak, aby výsledné schéma bylo přehledné, jako tomu je na obrázku č. 4.2. Modul bude mít na starosti vytvoření, umístění, nastavení a správné spojení uzlů Render Layers, File Output, Cryptomatte, Normalize a Denoise. v následujících sekcích popisují funkce jednotlivých uzlů.

## Kompoziční uzel Render Layers

Jak lze vidět na obrázku č. 4.3, tento uzel nabízí výstupy z renderovacího procesu. Výstupy jsou kompoziční vrstvy, které se mění na základě uživatelských nastavení v okně „Output Properties“. V uzlu je možné nastavit scénu a render vrstvu, které s renderovací proces týká.

Výstupy z uzlu se budou měnit na základě uživatelských nastavení a s tím se bude měnit zapojení vytvořené zásuvným modulem. V uzlu je možné nastavit render vrstvu, to jsem využil při renderování kombinované vrstvy ostatních render vrstev. Uživatel má možnost zaškrtnout, že pro každou vrstvu s povoleným renderováním se vygeneruje uzel Render Layers, nastaví si render vrstvu a spojí výstup Image s (kombinovaná vrstva) s adekvátním vstupem v uzlu File Output.



■ Obrázek 4.3 Kompoziční uzly

První výjimkou jsou výstupy `CryptomatteObject`, které v modulu používat nebudu. Slouží jako vstupní data pro uzel `Cryptomatte` verze 1 (v modulu využívám uzel s verzí 2). Druhou výjimkou jsou výstupy `Denoise`. Kompoziční vrstvy `Denoise Albedo` a `Denoise Normal` předáme společně s vybranou kompoziční vrstvou do uzlu `Denoise`.

## Kompoziční uzel `Denoise`

Uzel `Denoise` nad kompoziční vrstvou provede odstranění šumu s pomocí knihovny `OpenImage-Denoise` od Intelu [16]. Kombinovaná vrstva (na obrázku č. 4.3 se tato kompoziční vrstva jmenuje „Image“) odšuměním projde automaticky při renderování (Pokud tak uživatel specifikoval v nastavení renderování), takže může být z tohoto kroku vynechána.

## Kompoziční uzel `File Output`

Tento uzel automaticky exportuje předaná data do cílové složky. Na obrázku č. 4.3 lze vidět, že každý socket má své jméno a výstupní formát. Tyto data i počet socketů je možné upravovat. Na obrázku č. 4.3 lze vidět, že uzel `Render Pass Output` (File Output) přijímá již odšuměné kompoziční vrstvy a ukládá je do cílové složky. Cílovou složku nastaví modul automaticky podle preferencí uživatele v nastavení zásuvného modulu. Aby bylo možné přidat jménům prefix, modul smaže všechny sokety a pro každého kandidáta na export vytvoří nový vstupní socket a přejmenuje ho.

## Kompoziční uzel `Cryptomatte`

Uzel `Cryptomatte`, který lze vidět na obrázku č. 4.3, vytváří pixelové masky objektů. Na vstupu kombinovanou vrstvu, která obsahuje `cryptomatte` data (lze nastavit v okně „Output Properties“). Následuje nastavení zdroje `cryptomatte` dat, relevantní scény, `cryptomatte` vrstvy a vstupní pole `Matte ID`.

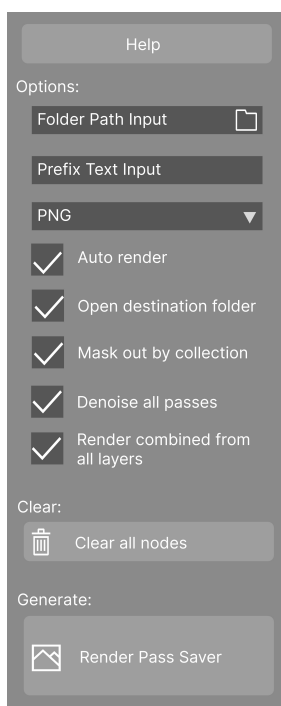
Při řešení maskování používám uzel „`Cryptomatte`“ verze 2. Nastavení aktuálního renderu jako zdroje `Cryptomatte` dat a aktuální render vrstvy jako `cryptomatte` vrstvy je vyhovující ve výchozím stavu, takže není třeba měnit v modulu. V mém návrhu budu potřebovat pouze výstup `Image`.

Do vstupního pole „`Matte ID`“ patří identifikátor (jméno) objektů, které chci maskovat. Vstupní typ je python list, takže můžu specifikovat množinu objektů a vytvoří se jejich společná

maska. Tento kompičnický uzel má tři výstupy. Prvním je Image, kombinovaná vrstva s aplikovanou maskou (vyříznuté objekty z kombinované vrstvy s transparentním pozadím). Druhým výstupem je Matte, černobílá maska, použitá ke generaci výstupu Image. Třetím výstupem je Pick, barevná maska s černým pozadím. Na ní lze provádět výběr podle barvy.

Pokud si uživatel zvolí „Mask out by collection“, modul vytvoří Cryptomatte uzel pro každou kolekci, která obsahuje viditelné objekty, a naplní Matte ID množinou objektů v kolekci. Výstupy pak propojí do vstupních socketů uzlu Render Pass Output a pojmenuje je podle jmen kolekcí (navíc se společným prefixem).

## Uživatelské rozhraní



■ **Obrázek 4.4** Návrh uživatelského rozhraní pro Blender modul

Zásuvný modul pro aplikaci Blender bude přístupný v okně Kompozitoru v liště po pravé straně.

Jeho grafické rozhraní se bude skládat z jednoduchého panelu nazvaného „Auto Render Pass Saver“. Tento panel bude obsahovat uživatelská nastavení a tři tlačítka. První tlačítko vyobrazené v návrhu na obrázku č. 4.4 se jmenuje „Help“ a jeho stisknutí otevře vyskakovací okno, ve kterém bude vypsaný uživatelský manuál. V uživatelském manuálu se píše, jak zásuvný modul používat, co nastavit před jeho použitím a jaké možnosti nabízí. Stručné informace o jednotlivých prvcích jsou také k dispozici při pohybu myši na pozici prvku.

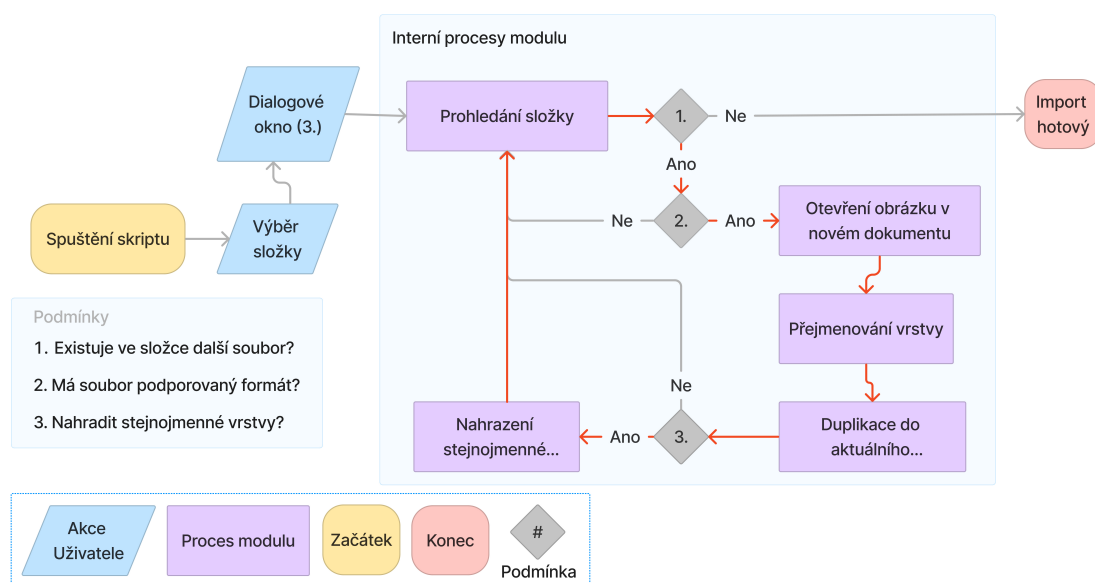
Malý rozdělovač specifikuje že dále následuje nastavení modulu. První je nastavení cílové složky. To lze provést textově nebo průchodem dialogu po kliknutí na ikonu adresáře. Pod tím je textové pole pro nastavení společného prefixu. Dále má uživatel možnost povolit nebo zakázat dostupné funkcionality modulu.

Jakmile uživatel nastaví modul, má k dispozici tlačítko „Clear all nodes“. Zvolil jsem pro toto tlačítko menší velikost, protože jeho použití je pouze situační. V průchodu modulem může nastat situace, kdy uživatel přivítá možnost smazat všechny uzly v Kompozitoru, ale nejedná se

o esenciální prvek modulu.

Tlačítko na spodku uživatelského rozhraní se jmenuje „Render Pass Saver“. Toto tlačítko je vizuálně větší, protože je důležité. Při pohledu na rozhraní by mělo být intuitivní, jak modul spustit.

## 4.2 Zásuvný modul pro Photoshop



■ **Obrázek 4.5** Diagram aktivít zásuvného modulu pro Photoshop

Zásuvný modul pro Photoshop bude importovat soubory (jako vrstvy) do aktuálního dokumentu z vybraného adresáře. Tento proces nevyžaduje žádné složité grafické rozhraní. Na obrázku č. 4.5 je diagram aktivít, podle kterého zásuvný modul bude postupovat. Uživatel bude mít možnost nahradit stejnojmenné vrstvy. Přepsání vrstev se stejným jménem pomáhá zkrátit proces výměny vrstev při opětovném renderování a zároveň uživatele nabádá, aby s vyrenderovanými vrstvami pracoval nedestructivně (veškeré změny prováděl na nových vrstvách a ty importované nechal bez změn). Na diagramu je vyznačený červeně cyklus, který se provede pro každý soubor ve vybrané složce.

## Nalezení kompozičních vrstev

Jako první je třeba, aby uživatel specifikoval adresář s exportovanými kompozičními vrstvami. Skript spustí adresářový dialog a uživatel na lokálním systému najde a potvrdí adresář. V případě, kdy se skriptu nepodaří specifikovaný adresář najít nebo adresář neobsahuje žádné soubory s podporovaným formátem, skript vypíše chybovou hlášku a skončí. Podporované obrázkové formáty jsou: PNG, JPG, BMP, TIF a GIF. Modul se nestará o selekci kompozičních vrstev podle nastaveného prefixu, proto je rozumné ukládat výstupy z Blenderu do prázdného adresáře.

## Spuštění modulu bez otevřeného dokumentu

Pro splnění požadavku F2.2, modul si musí poradit, pokud je volán bez otevřeného dokumentu. V takovém případě modul otevře první kompoziční vrstvu z vybrané složky a všechny ostatní

vrstvy se importují do dokumentu s první importovanou vrstvou. Tento způsob jsem zvolil, aby se uživatel nemusel starat o rozměry nového dokumentu. Všechny kompoziční vrstvy mají vždy stejné rozměry jako finální obrázek, takže import jedné z vrstev automaticky vytvoří dokument se správnými rozměry. Žádné přeskálování ani zarovnání vrstev není třeba. Po importu všech kompozičních vrstev z vybraného adresáře se v tomto případě automaticky spustí dialog „Uložit jako...“.

## Import kompozičních vrstev

Jakmile modul otevře složku, prohledá všechny soubory a s pomocí regulárních výrazů uloží si jména všech podporovaných souborů. Poté po jednom otevře soubory jako nové dokumenty, přejmenuje hlavní vrstvu, aby převzala jméno souboru bez označení formátu a tuto vrstvu duplikuje na vršek původního dokumentu. Následně dokument s otevřeným souborem zavře bez uložení. Modul nepracuje v pozadí, naopak všechny změny jsou viditelné skrze uživatelské rozhraní Photoshopu, ale jsou v rychlém sledu, takže z pohledu uživatele modul skáče z jednoho okna do druhého.

## Náhrada kompozičních vrstev

Funkční požadavek F2.3 specifikoval možnost nahradit již existující kompoziční vrstvy v aktivním dokumentu nově importovanými vrstvami se stejným jménem. Poté co modul správně pojmenuje vrstvu obsahující importovaný obrázek, prohledá všechny vrstvy aktivního dokumentu a pokud najde shodu, smaže stejnojmennou vrstvu z aktivního dokumentu. Následně modul duplikuje importovanou vrstvu do aktivního dokumentu na místo vrstvy smazané.



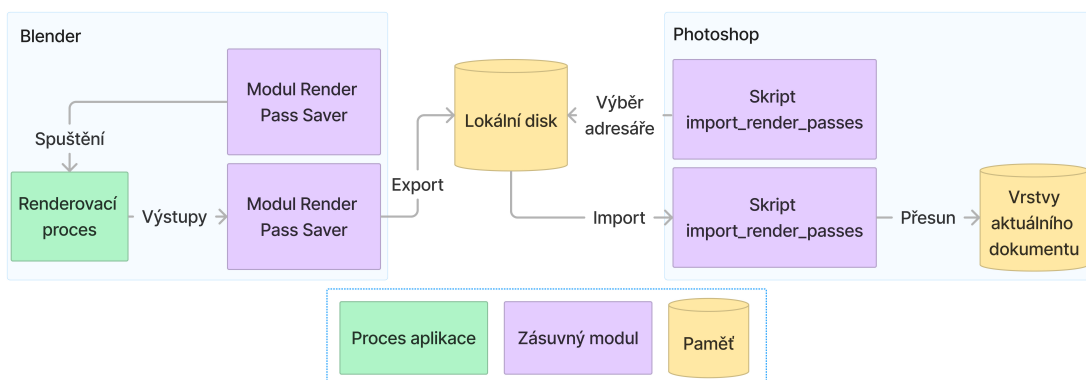
# Implementace

V následující kapitole popisují, jak zásuvné moduly nainstalovat a používat. Dále popisují některé komplikace, na které jsem narazil při implementaci, a nakonec specifikuji splněné a nesplněné požadavky.

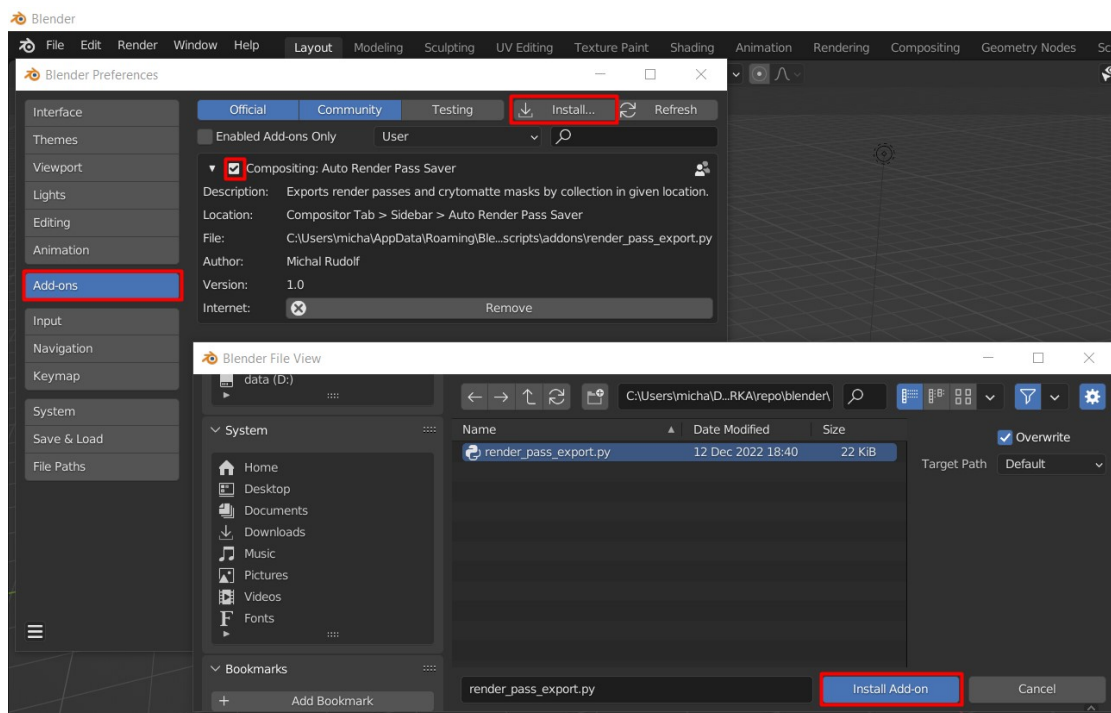
### 5.1 Schéma nasazení

Zásuvné moduly jsou vyvinuty, aby zjednodušily a zrychlily přesun kompozičních dat mezi programy Blender a Photoshop a zároveň, aby nabídly lepší způsob maskování objektů. Schéma zapojení lze vidět na obrázku č. 5.1. Modul pro Blender lze využít, jakmile je scéna hotová a čeká se na spuštění renderovacího procesu. V tuto chvíli uživatel specifikuje nastavení modulu a spustí modul tlačítkem „Render Pass Saver“. Podle diagramu na obrázku č. 4.1, se na základě uživatelských nastavení provedou interní operace modulu a všechny kompoziční vrstvy se exportují do určeného adresáře na lokálním disku.

Photoshop skript se spouští přímo z aplikace. Poté, co uživatel specifikuje adresář s výstupy renderování, skript prohledá adresář na lokálním disku a importuje všechny obrázkové soubory. Následně ze všech importovaných obrázků udělá nové vrstvy aktuálního dokumentu.



■ **Obrázek 5.1** Uživatelův postup práce s moduly



■ Obrázek 5.2 Instalace modulu v aplikaci Blender (verze 3.1.2)

## 5.2 Instalační příručka

Zásuvné moduly jsou dostupné v přiloženém elektronickém médiu nebo na git repozitáři pod adresou <https://gitlab.fit.cvut.cz/rudolmi6/bakalarni-prace>.

### Instalace modulu pro Blender

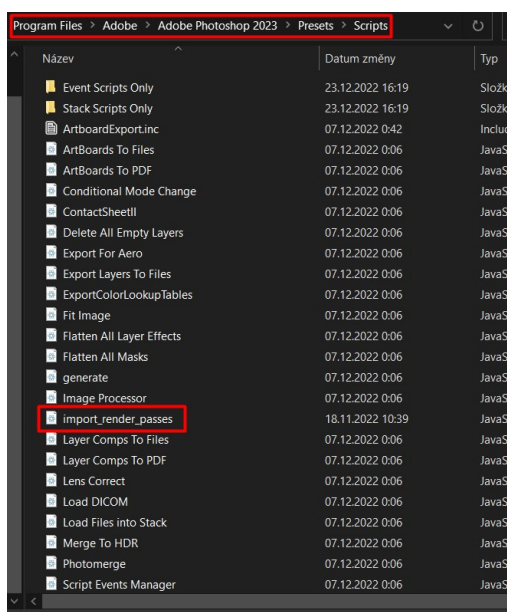
Modul nainstaluji v aplikaci na cestě `Edit > Preferences... > Add-ons > Install...`, otevřu souborový dialog, ve kterém vyberu soubor `render_pass_saver.py` z přiloženého elektronického média. Jakmile se zásuvný modul nahraje, objeví se v nabídce. Povolím jej zaškrtnutím políčka. Všechny popsané operace lze vidět zvýrazněny na obrázku č. 5.2.

Zásuvný modul se objeví v okně Kompozitoru, v postranním panelu pod jménem „Auto Render Pass Saver“.

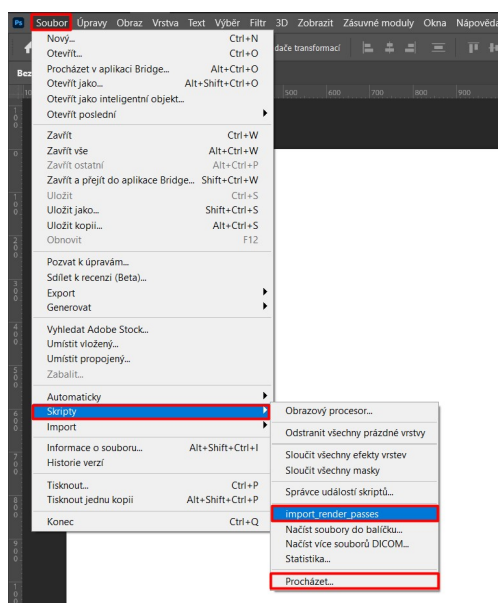
### Instalace skriptu pro Photoshop

První metoda umožňuje spustit skript jednorázově. V aplikaci Photoshop spustíme dialog na cestě `Soubor > Skripty > Procházet...` a dialogu vybereme soubor `import_render_passes.jsx` (viz. obrázek č. 5.4). Tímto se skript automaticky spustí.

Druhý způsob umožňuje uložení skriptu do stálé nabídky. Na lokálním disku otevřu instalační složku Photoshopu (Většinou na cestě `C:\Program Files\Adobe\Adobe Photoshop...`). V instalační složce dále pokračuji do adresáře `Presets\Scripts` a do něj umístím skript `import_render_passes.jsx` (viz. obrázek č. 5.3). Nyní je třeba restartovat Photoshop. Skript bude nadále k dispozici v aplikaci na cestě `Soubor > Skripty > import_render_passes` (viz. obrázek č. 5.4).



■ Obrázek 5.3 Uložení skriptu do Presets



■ Obrázek 5.4 Volání skriptu (PS verze 24.1.0)

### 5.3 Uživatelská příručka

Moduly se používají v sérii po sobě. Nejdříve export z Blenderu s pomocí zásuvného modulu Render Pass Saver a následně import do Photoshopu s pomocí skriptu import\_render\_passes.

#### Před použitím modulů

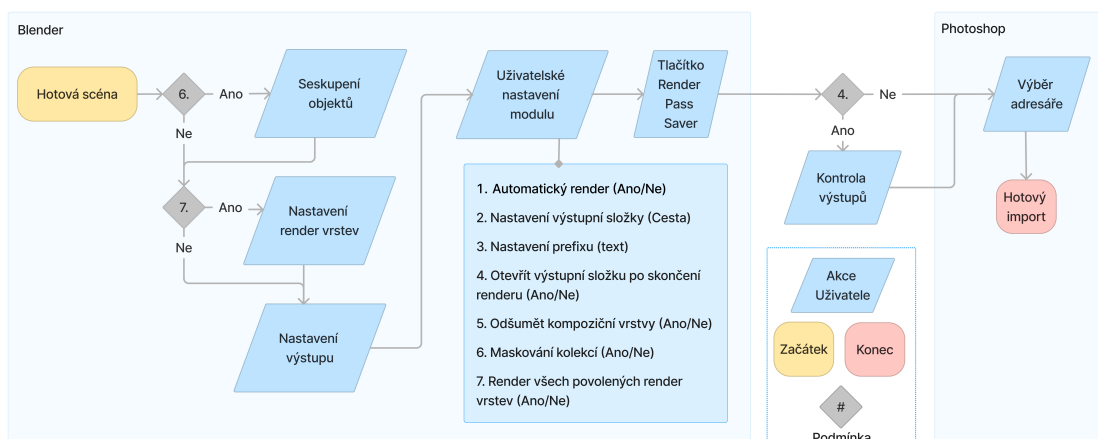
Modul pro Blender počítá s tím, že uživatel dokončil tvorbu a nastavení scény. Nastavení, která je potřeba provést před použitím modulu se liší v závislosti na uživatelských preferencích, jak můžete vidět na obrázku č. 5.5. Seskupení objektů do kolekcí nebo nastavení ostatních render vrstev může uživatel vynechat, pokud tyto funkcionality modulu nechce využít. Nastavení výstupů je však povinné. Mezi to patří nastavení, jaké kompoziční vrstvy by se měly vyrenderovat.

#### Renderování ostatních render vrstev

Pokud chce uživatel vyrenderovat i ostatní render vrstvy, musí před použitím modulu přepnout do render vrstvy, ze které chce exportovat kompoziční vrstvy. Dále pro všechny ostatní render vrstvy, je třeba povolit možnost „Use for rendering“. Na základě tohoto nastavení modul pozná, které vrstvy vyrenderovat a které ne. Z ostatních vrstev se exportuje pouze kombinovaná vrstva.

#### Odstranění šumu

Pokud chce uživatel odstranit šum, je třeba v hlavní render vrstvě povolit Denoising v okně „Render Properties“ a nechat nastavenou výchozí metodu (OpenImageDenoise - Albedo and Normal passes). Dále uživatel povolí renderování kompoziční vrstvy „Denoise Data“. Modul je ošetřen proti špatnému nastavení a oznámí chybu.



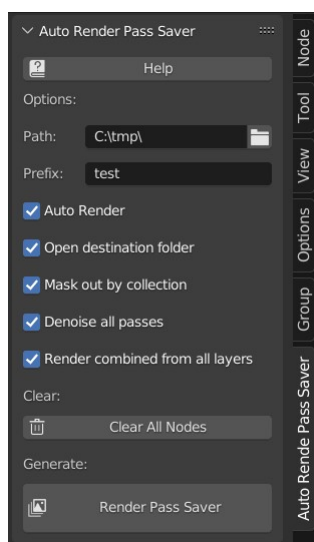
■ Obrázek 5.5 Uživatelský postup práce s moduly

## Maskování podle kolekcí

Pokud chce uživatel využít funkcionalitu „Mask out by collection“ je třeba, aby si zorganizoval Průzkumník scény (Scene Collection). Seskupí jednotlivé objekty, které chce na společné separátní vrstvě. Skupiny, které obsahují pouze nehmotné objekty (světla a kamery) se přeskočí. Dále je pro správného fungování funkcionality nutné povolit kompoziční vrstvu „CryptomatteObject“, ale modul je ošetřen a oznámí chybu, pokud se tak nestane.

## Nastavení modulu Render Pass Saver

V grafickém rozhraní modulu pro Blender je tlačítko „Help“ (viz. obrázek č. 5.6), které zobrazí celý uživatelský manuál v anglickém jazyce. Tlačítkem „Clear“ lze smazat všechny uzly v Kompozitoru.



■ Obrázek 5.6 Uživatelské rozhraní modulu pro Blender

- Export folder Path: Nastavení cílového adresáře.
- Prefix: Společný prefix jména každého výstupu (počáteční část jména oddělena „-“). Je doporučeno každou várku pojmenovat jinak, aby se výstupy nepřemazaly.
- Auto-render: Modul po provedení interních procesů automaticky zahájí renderovací proces.
- Open destination folder: Modul po skončení renderovacího procesu automaticky otevře cílový adresáře.
- Mask out by collection: Každá kolekce (neprázdná skupina objektů, světla a kamery se přeskakují).
- Denoise all passes: Modul odstraní šum každé kompoziční vrstvy.
- Render other layers: Modul spustí renderovací proces i všem ostatním povoleným render vrstvám.

Po nastavení modulu stačí stisknout tlačítko „Render Pass Saver“ a modul se spustí.

## Skript pro Photoshop

Skript lze spustit s otevřeným dokumentem nebo bez otevřeného dokumentu. V prvním případě skript importuje obrázkové soubory z vybraného adresáře jako nové vrstvy aktuálního dokumentu. Ve druhém případě bez otevřeného dokumentu skript automaticky vytvoří nový dokument, importuje do něj obrázkové soubory z vybraného adresáře a po dokončení spustí dialog „Uložit jako...“. Skript automaticky pojmenuje vrstvy po jednotlivých importovaných souborech.

### 5.4 Programátorský příručka

Ve zdrojových kódech jsou všechny funkce zdokumentovány podle příruček JSDoc (pro JavaScript) a PEP 257 (pro Python).

## Modul pro Blender

Zásuvný modul je rozdělen do několika částí:

- Metadata
- Třída `RenderPassSaverProperties`
- Panel `RenderPassSaver_PT_Panel`
- Operátor `UserManualButton`
- Operátor `ClearNodesButton`
- Operátor `RenderPassButton`

Zásuvný modul začíná inicializací metadat, které jsem popsal v kapitole Návrh.

Třída `RenderPassSaverProperties` slouží jako kontejner pro uživatelské preference. Jedná se o modul přidanou členskou proměnnou třídy `Scene`. Programátor může příkazem `bpy.context.scene.RenderPassSaverProperties` zpřístupnit třídu patřící aktivní scéně.

Panel `RenderPassSaver_PT_Panel` je grafickým rozhraním modulu. Nachází se na něm všechny nastavitelné i funkční prvky.

Operátory `UserManualButton` a `ClearNodesButton` jsou přístupné v grafickém rozhraní, jako tlačítka „Help“ a „Clear All Nodes“.

Operátor `RenderPassButton` je složitější, obsahuje velkou část implementačního řešení modulu. Operátor je přístupný v grafickém rozhraní jako tlačítko „Render Pass Saver“ a jeho stisknutí spustí specifické procesy povolené v uživatelském nastavení modulu. Interní procesy se provádí z metody `execute()`, která postupně vygeneruje a propojí potřebné kompoziční uzly.

## Skript pro Photoshop

Photoshop podporuje volání skriptů v jazyce JavaScript. Adobe nabízí více způsobů vývoje zásuvných modulů a skriptů a většina z nich nemá snadně dohledatelnou dokumentaci. K implementaci skriptu jsem používal dokumentaci na webu <https://theviaxx.github.io/photoshop-docs/index.html>.

Na začátek skriptu jsem umístil příkaz `#target photoshop`, specifikuje, pro jakou aplikaci je skript určený. Uživatel může spustit skript dvojklikem a skript se automaticky spustí ve Photoshopu. Pokud aplikace na systému v tu chvíli neběží, skript jí spustí.

Při programování skriptu lze k aplikaci přistupovat přes strukturu `app`, která v sobě zapouzdřuje rozhraní Photoshopu. Přístup k aktivnímu dokumentu lze získat zavoláním `app.activeDocument`. Přes aktivní dokument dále přistupuji k jeho vrstvám nebo funkcím, které nabízí Photoshop při ovládání přes grafické rozhraní.

Skript spouští hlavní proceduru `importRenderPasses` a chytá případnou výjimku. Tato funkce volá adresářový dialog, následně extrakci podporovaných souborů z vybraného adresáře, a nakonec import po jednom souboru.

## 5.5 Komplikace a jejich řešení

### Automatické spuštění render engineu

Při implementaci jsem považoval za důležité, aby zásuvný modul uměl automaticky spustit renderovací proces, ale narazil jsem na problém. Renderovací proces se spustí příkazem `bpy.ops.render.render()`, ale pokud tuto funkci spustíme ve funkci `execute()` tlačítka `Render Pass Saver`, pak se renderovací proces volá ze stejného vlákna jako grafické rozhraní. Renderovací proces se spustí a grafické rozhraní zamrzne, dokud renderovací proces neskončí. Dalším nechtěným vedlejším efektem je, že pokud se renderovací proces spustí tímto způsobem, uživatel nemá možnost jej zrušit a nevidí žádné průběžné indikátory, jak dlouho už proces běží a v kolika je procentech. Renderovací příkaz se proto musí volat ze separátního vlákna.

```

1 # Funkce execute() tlačítka Render Pass Saver, které spustí renderovací proces
2 # Registrace callback handleru
3 bpy.app.handlers.render_pre.append(self.PreRender)
4 bpy.app.handlers.render_post.append(self.PostRender)
5 bpy.app.handlers.render_cancel.append(self.CancelledRender)
6 # Nastavení časovace - po kazde pul vterine posle udalost skrze modal
7 self.ui_timer = bpy.context.window_manager.event_timer_add(0.5, window =
  context.window)
8 # Registrace modalu
9 bpy.context.window_manager.modal_handler_add(self)
10 return {"RUNNING_MODAL"}
```

#### ■ Výpis kódu 5.1 Registrace modalu

Volání v separátním vlákně jsem docílil registrováním render callbacků a použitím modalu. Nastavil jsem časovač s periodou půl vteřiny. Tento časovač, periodicky posílá v jiném vlákně událost do modalu. render callbacky jsem využil k aktualizaci informace, zda renderovací proces již skončil. Jakmile callback volající se po konci renderu nastaví, že renderovací proces skončil

nebo byl zastaven uživatelem, modal do půl vteřiny všechny callbacky a handlery odhlásí. Tímto způsobem se grafické rozhraní nezasekne a spustí se klasické renderovací okno.

## Renderování clown passu

Funkční požadavek F1.3 specifikuje, že bude možné automaticky renderovat clown pass. Pro splnění tohoto úkolu je třeba přepnout na Workbench render engine, v záložce „Render Properties“ nastavit světlo na „Flat“ a barvu na „Random“. Při implementaci jsem však zjistil, že měnit nastavení jiného kontextu je s pomocí Blender Python API je velmi obtížné. Zásuvný modul pracuje v kontextu Kompozitoru a nepodařilo se mi z tohoto kontextu získat přístup k nastavení „Render Properties“. Rozhodl jsem se tedy tento požadavek přeskočit.

## Přejmenování výstupních souborů

Při exportu s použitím kompozičních uzlů se k výstupu automaticky přiloží čtyřmístné číslo snímku (pro renderování animací). To je v mém případě nechtěné a na fóru jsem zjistil, že v Blender Python API, při generaci výstupů, nejde tento proces zvrátit [17]. Nakonec jsem alespoň změnil počet cifer na jednu znakem „#“ na konci každého jména.

## Nahrazení stejnojmenných vrstev

Podle funkčního požadavku F2.3 měl zásuvný modul pro Photoshop nabízet možnost, při importu nahradit stávající vrstvy importovanými vrstvami, pokud mají totožné jméno. Tuto funkcionalitu jsem však nestihl implementovat.

## 5.6 Kontrola splnění funkčních požadavků

Výsledné zásuvné moduly splňují 13 z 16 definovaných požadavků. Splněné a nesplněné požadavky lze vidět na tabulce č. 5.1.

■ **Tabulka 5.1** Splnění požadavků

Požadavek	Splněno (X)
F1.1	X
F1.2	X
F1.3	-
F1.4	X
F1.5	X
F1.6	X
F1.7	X
F1.8	X
F1.9	X
F1.10	-
F1.11	X
NF1.1	X
F2.1	X
F2.2	X
F2.3	-
NF2.1	X





*V následující kapitole popisují uživatelské testování vytvořených zásuvných modulů, popisují tvorbu testů, výsledky testů, a nakonec místa pro zlepšení.*

Implementované zásuvné moduly byly vytvořeny pro cílovou skupinu, za účelem zrychlit a zjednodušit přesun dat mezi programy. Právě tyto procesy budou předmětem uživatelského testování.

Prvním krokem bylo vytvořit veřejný repozitář se zásuvnými moduly, s návodem, jak je nainstalovat a použít. Pro tyto účely jsem vytvořil repozitář „Blender-Photoshop workflow plug-ins“ na serveru gitlab.fit.cvut (<https://gitlab.fit.cvut.cz/rudolmi6/bakalarni-prace>). Tento repozitář jsem rozesílal respondentům z průzkumu. Profesionální umělci jsou často velmi zaneprázdnění, a proto nejsou vhodnými kandidáty pro klasické uživatelské testování s přesnými pokyny a scénáři. Na druhou stranu vytvořené zásuvné moduly by měly sloužit právě jim, takže jsem vytvořil testovací scénář, který dává umělcům volnost využít moduly na libovolný projekt a ukázat tak užitečnost modulů.

### 6.1 Scénář uživatelského testu pro umělce

1. Otevřete si libovolnou scénu. V záložce „View Layer Properties“ specifikujte kompoziční vrstvy, které běžně používáte při práci. Upravte nastavení renderování, jaké běžně používáte při své práci.
2. Přepněte do Kompozitoru, otevřete zásuvný modul Auto Render Pass Saver. V zásuvném modulu nastavte cílovou složku exportu a povolte veškeré dostupné možnosti. Stiskněte tlačítko „Render Pass Saver“.
3. Po dokončení renderování otevřete Photoshop a pomocí skriptu `import_render_passes.jsx` importujte všechny soubory z adresáře s kompozičními vrstvami.

Užitečnost zásuvného modulu jsem definoval třemi tezemi. Pokud umělec považuje alespň jeden z nich za pravdivý, zásuvný modul je pro ně užitečný.

- Zásuvné moduly šetří čas nebo úsilí při práci s programy.
- Zásuvné moduly nabízí chtěnou funkcionalitu, kterou pracovní proces do teď neobsahoval.
- Zásuvné moduly zjednodušují pracovní proces s programy.

## 6.2 Výsledky testů

Podařilo se mi získat zpětnou vazbu od 3 profesionálních umělců, kteří se zúčastnili mého průzkumu (dále testeři). Každý z testerů bez problému nainstaloval moduly podle návodu a vyzkoušel jejich funkce na projektu své volby.

Tester č. 1 potvrdil, že oba zásuvné moduly šetří čas při přesunu dat mezi programy. Zároveň zmínil, že generování cryptomatte masek se mu hodí do jeho pracovního postupu. Dále nabídl pro oba moduly nápady pro zlepšení, které popíše v sekci Prostor pro zlepšení. Nakonec vyjádřil vůli sdílet zásuvné moduly svým kolegům umělcům. Tester č. 2 potvrdil, že automatické odstranění šumu a generování masek jsou užitečné funkcionality, které bude využívat na budoucích projektech. Tester č. 3 potvrdil, že zásuvné moduly jsou užitečné, ale ne nutně v každém projektu, protože využití více kompozičních vrstev není vhodné pro každý projekt.

## Rychlost Photoshop skriptu

Skript, který jsem popsal v sekci Současná řešení (`import_from_folder.jsx`), využívá podobný princip, ale vykonává některé zbytečné operace. Jeho rychlost se pohybuje kolem jedné importované vrstvy za 1,3 vteřiny. Můj skript se pohybuje lehce pod hranicí jednoho importovaného obrázku za vteřinu.

## 6.3 Prostor pro zlepšení

Jeden z testerů poukázal na nechtěné chování zásuvného modulu pro Blender: Pokud jsou pro export využity stejná jména i cílový adresář, výsledné výstupy přepíše již existující data. Aby se tento problém vyřešil, modul by musel prohledat cílový adresář a při shodě upozornit uživatele dialogem.

Další bod zpětné vazby se týká skriptu pro Photoshop. Při otevření adresářového dialogu si skript nepamatuje cestu k naposled vybranému adresáři. Tato úprava by uživateli ušetřila čas při hledání adresáře s kompozičními vrstvami na lokálním disku.

Při tvorbě zásuvných modulů je velmi důležitou částí proces vydání. Čím více lidí se dozví o výsledných modulech, tím více potenciálních uživatelů budou mít. V této práci jsem tuto část nestihl a pracoval jsem pouze s členy průzkumu, ale dalším krokem by bylo moduly sdílet na uživatelských fórech a trzích jako je např. BlenderMarket nebo Adobe Marketplace, aby byly snadně dohledatelné.

## Kapitola 7

# Závěr

*V následující kapitole shrnuji celou bakalářskou práci a hodnotím, zda byly jednotlivé požadavky zadání splněny.*

V této práci jsem měl za cíl identifikovat problémy sužující umělce při práci s programy Blender a Photoshop a tyto problémy řešit pomocí zásuvných modulů. V první kapitole jsem vysvětlil pojmy spojené se digitální malbou, zásuvnými moduly a specifika pracovního postupu digitální malby s programy Blender a Photoshop.

V další kapitole jsem se věnoval průzkumu mezi profesionály v oborech digitální malby. Popsal jsem zkušenosti s průzkumem a provedl analýzu, ve které rozebírám konkrétní odpovědi.

V následující kapitole jsem vyzkoušel různé API knihovny, s jejichž pomocí lze tvořit zásuvné moduly pro Blender a Photoshop. Dále jsem popsal příklady současných řešení modulů věnující podobné problematice. V další sekci jsem specifikoval problémy pracovního postupu Blender-Photoshop identifikované v průzkumu a následně jsem definoval funkční a nefunkční požadavky pro zásuvné moduly řešící tyto problémy.

V další kapitole jsem navrhl jeden zásuvný modul pro Blender a jeden pro Photoshop tak, aby splnily všechny definované požadavky. Podařilo se mi implementovat zásuvné moduly, které splňují 13 z 16 požadavků.

V kapitole „Implementace“ jsem popsal nasazení zásuvných modulů, instalační, uživatelskou a programátorskou příručku a nakonec některé implementační překážky, které jsem musel překonat.

Na závěr jsem podrobil zásuvné moduly uživatelským testům, které prováděli účastníci mého průzkumu.

V práci jsem splnil všechny cíle, které byly zadány. Dalším postupem práce by bylo vydání zásuvných modulů na komunitních serverech, tak aby byly volně přístupné co největšímu počtu uživatelů. Výsledkem práce jsou zásuvné moduly, které zjednodušují a zrychlují přesun kompozičních dat mezi programy a dávají tak umělcům více prostoru pro kreativní činnost.



# Bibliografie

1. WANG1, Vivian; WANG, Dali. The Impact of the Increasing Popularity of Digital Art on the Current Job Market for Artists. *Art and Design Review* [online]. 2021, roč. 9, č. 3 [cit. 2022-01-05]. ISSN 242-253. Dostupné z DOI: 10.4236/adr.2021.93019.
2. KONTKANEN, Iida. *Using 3D Techniques in Video Game Concept Art* [online]. Tampere, Finsko, 2021 [cit. 2022-01-05]. Dostupné z: [https://www.theseus.fi/bitstream/handle/10024/501432/Iida\\_Kontkanen.pdf?sequence=5&isAllowed=y](https://www.theseus.fi/bitstream/handle/10024/501432/Iida_Kontkanen.pdf?sequence=5&isAllowed=y). Bakalářská práce. Tampere University of Applied Sciences.
3. RENDERGUIDE. *The Impact of the Increasing Popularity of Digital Art on the Current Job Market for Artists* [online]. RenderGuide.com, 2021. [cit. 2022-01-05]. Dostupné z: <https://renderguide.com/blender-composition-tutorial/>.
4. KURENKOV, Artem. *Balancing 2D & 3D Techniques in Concept Art* [online]. 80lv, 2020-04-06. [cit. 2022-01-05]. Dostupné z: <https://80.lv/articles/balancing-2d-3d-techniques-in-concept-art/>.
5. FRAM, Lauren. *Best Concept Art Tools for Indie Game Designers* [online]. G2.com, 2019-03-18. [cit. 2022-01-05]. Dostupné z: <https://learn.g2.com/best-concept-art-tools>.
6. ROWLANDS, Chris; WILSON, Mark. *Best photo editors of 2022* [online]. TechRada.com, 2022. [cit. 2022-01-05]. Dostupné z: <https://www.techradar.com/news/best-photo-editor>.
7. MORRIS, Trevor. *Scripting Photoshop, Part 1 — An Introduction* [online]. Trevor Morris Photographics, 2014. [cit. 2022-01-05]. Dostupné z: <https://morris-photographics.com/photoshop/tutorials/scripting1.html>.
8. SIDDI, Francesco. *Blender by the Numbers – 2020* [online]. Blender Foundation, 2021-07-07. [cit. 2022-01-05]. Dostupné z: <https://www.blender.org/news/blender-by-the-numbers-2020/>.
9. WALKER, Tom. *20 Most Popular Open Source Software Ever* [online]. Trip Wire Magazine, 2017-01-15. [cit. 2022-01-05]. Dostupné z: <https://www.tripwiremagazine.com/20-most-popular-open-source-software-ever-2/>.
10. PLURALSIGHT. *Best Concept Art Software — Free Tools Included* [online]. Pluralsight, 2014-05-22. [cit. 2022-01-05]. Dostupné z: <https://www.pluralsight.com/blog/film-games/vfx-file-formats-you-need-to-know>.
11. MA, Shawn. *SteamPunk Scene* [online]. Artstation, 2019-11-03. [cit. 2022-01-05]. Dostupné z: <https://www.artstation.com/artwork/ba4ovk>.
12. *Písemná korespondence s profesionály v oboru digitální malby*. Artstation, 2022. Online korespondence se nachází v příloze.

13. CRAWFORD, Abby. *600,000 Free 3D Models on Sketchfab* [online]. Sketchfab, 2021-03-04. [cit. 2022-01-05]. Dostupné z: <https://sketchfab.com/blogs/community/600000-free-3d-models-on-sketchfab/>.
14. FOUNDATION, Blender. *Quickstart - Blender Python API* [online]. Blender Foundation, 2022-12-14. [cit. 2022-01-05]. Dostupné z: [https://docs.blender.org/api/current/info\\_quickstart.html](https://docs.blender.org/api/current/info_quickstart.html).
15. LIANG, Siew Yi. *Photoshop C++ Recipes* [online]. 2020-02-06. [cit. 2022-01-05]. Dostupné z: [https://sonickt.github.io/ps\\_cpp\\_recipes/#](https://sonickt.github.io/ps_cpp_recipes/#).
16. CORPORATION, Intel. *Open Image Denoise* [online]. Intel Corporation, 2019-01-29. [cit. 2022-01-05]. Dostupné z: <https://www.openimagedenoise.org/>.
17. EINAR. *Animation - How do you remove frames in Blender?* [Online]. Blender Stack Exchange, 2014-08-28. [cit. 2022-01-05]. Dostupné z: <https://blender.stackexchange.com/questions/14434/how-do-you-remove-frames-in-blender>.

# Obsah přiloženého média

readme.txt.....	stručný popis obsahu média
API_tests	
├ SDK_hello_world.....	testovací modul Photoshop SDK API
├ UXP_hello_world.....	testovací modul Adobe UXP API
└ blender_hello_world.py.....	testovací modul Blender Python API
text	
├ thesis.pdf.....	text práce ve formátu PDF
└ src.....	zdrojový kód práce v L <sup>A</sup> T <sub>E</sub> X
plugins	
├ import_render_passes.jsx.....	zásuvný modul pro Photoshop
└ render_pass_export.py.....	zásuvný modul pro Blender
resources	
├ surveys.pdf.....	písemná korespondence s umělci
└ suggestions.pdf.....	návrhy na zásuvné moduly od respondenta č. 4