

Využití procedurálních přístupů ke generování virtuálních prostředí pro VFX

Ing. Jan Trubač

Diplomová práce
2021

 Univerzita Tomáše Bati ve Zlíně
Fakulta multimediálních komunikací

Univerzita Tomáše Bati ve Zlíně
Fakulta multimediálních komunikací
Ateliér Audiovize

Akademický rok: 2020/2021

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Ing. Jan Trubač**
Osobní číslo: **K19431**
Studijní program: **N8209 Teorie a praxe audiovizuální tvorby**
Studijní obor: **Audiovizuální tvorba – Vizuální efekty**
Forma studia: **Prezenční**
Téma práce: **1. Teoretická část:**
Využití procedurálních přístupů ke generování virtuálních prostředí pro VFX 2
Praktická část:
**Samostatné audiovizuální dílo založené na využití VFX v délce odpovídající ná-
mětu a náročnosti technologie.**

Forma zpracování diplomové práce: **Tištěná/elektronická**

Seznam doporučené literatury:

- OKUN, Jeffrey A. a Zwerman, Susan. The VES Handbook of Visual Effects: Industry Standard VFX Practices and Procedures. Focal Press, 2010. ISBN 978-0-240-81242-7
- MÜLLER, Pascal, Peter WONKA, Simon HAEGLER, Andreas ULMER a Luc VAN GOOL. Procedural modeling of buildings. ACM Transactions on Graphics [online]. 2006, 25(3), 614–623. Dostupné z: doi:10.1145/1141911.1141931
- KELLY, Tom a Peter WONKA. Interactive architectural modeling with procedural extrusions. ACM Transactions on Graphics [online]. 2011, 30(2), 1–15. Dostupné z: doi:10.1145/1944846.1944854
- CHEN, Guoning, Gregory ESCH, Peter WONKA, Pascal MÜLLER a Eugene ZHANG. Interactive procedural street modeling. ACM Transactions on Graphics [online]. 2008, 27(3), 1–10. Dostupné z: doi:10.1145/1360612.1360702
- KADNER, Noah. The Virtual Production Field Guide [online]. Epic Games, ©2019. Dostupné z: <https://www.unrealengine.com/en-US/vpfieldguide>
- MAGEE, Robert, Houdini Foundations for Film, TV & Gamedev. SideFX Software, 2018. ISBN 978-1-7753338-0-7. Dostupné také z: https://www.sidefx.com/media/uploads/tutorial/foundations_gdc2018/houdini_foundations.pdf

Vedoucí teoretické části: **ak. mal. Boris Masník**
Ateliér Audiovize

Vedoucí praktické části: **ak. mal. Boris Masník**
Ateliér Audiovize

Datum zadání diplomové práce: **2. prosince 2020**

Termín odevzdání diplomové práce: **21. května 2021**



L.S.

doc. Mgr. Irena Armutidisová
děkanka

MgA. Irena Kocí, Ph.D.
vedoucí ateliéru

Ve Zlíně dne 2. prosince 2020

PROHLÁŠENÍ AUTORA DIPLOMOVÉ PRÁCE

Beru na vědomí, že

- diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému a bude dostupná k nahlédnutí;
- na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- podle § 60 odst. 2 a 3 mohu užít své dílo – diplomovou práci – nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tj. k nekomerčnímu využití), nelze výsledky diplomové práce využít ke komerčním účelům;
- pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji, že:

- jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.

Ve Zlíně dne: 28. 7. 2021

Jméno a příjmení studenta: Jan Trubač

.....
podpis studenta

ABSTRAKT

Tato práce se zaměřuje na proces vytváření procedurálních systémů a jejich porovnání s ostatními metodami využívanými v oblasti vizuálních efektů. Dále pojednává o výhodách a možných úskalích vývoje a nabízí vhodná řešení problémů. Výstupem této práce je krátký film, který demonstruje konkrétní využití procedurálně generovaných prostředí vytvořených v nástrojích Houdini a Substance Designer. Je zde využito nejmodernějších trendů v oblasti VFX, které jsou spojeny s využitím real-time řešení. Konkrétně je celý film renderován v Unreal Engine. Aby byl zajištěn co nejrealističtější výsledek, je využito techniky real-time ray tracing, která simuluje složité chování reálného světla a vytváří tak věrohodnější obraz.

Klíčová slova: Proceduralní generování, prostředí, Houdini, Substance Designer, Unreal Engine

ABSTRACT

This thesis focuses on the process of creating procedural systems and their comparison with other methods used in the field of visual effects. It further discusses the advantages and possible pitfalls in the development and offers suitable solutions to the problems.

The output of this work is a short film that demonstrates the specific use of procedurally generated environments created in Houdini and Substance Designer. It utilizes the latest trends in the field of visual effects, which are associated with real-time solutions. In this case, the entire film is rendered with the Unreal Engine. State-of-the-art techniques, such as real-time ray tracing, are used to ensure the most realistic appearance of the final images. This technique simulates the complex behavior of real light and thus creates a more believable image.

Keywords: Procedural generation, environment, Houdini, Substance Designer, Unreal Engine

Chtěl bych na tomto místě poděkovat panu ak.mal. Borisi Masníkovi za cenné rady a konzultace nejen při vzniku této práce. Můj velký dík patří také rodině a přátelům za jejich neustálou podporu a trpělivost.

Prohlašuji, že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 PROCEDURÁLNÍ PŘÍSTUP	12
1.1 OBLASTI VYUŽITÍ.....	12
1.2 PROCEDURÁLNÍ TECHNIKY V KOMBINACI S REAL-TIME TECHNOLOGIEMI	15
1.3 VÝHODY PROCEDURÁLNÍCH SYSTÉMŮ	16
1.4 NEVÝHODY PROCEDURÁLNÍCH SYSTÉMŮ	17
2 POROVNÁNÍ PROCEDURÁLNÍHO PŘÍSTUPU S OSTATNÍMI TECHNIKAMI ..	17
2.1 POROVNÁNÍ PŘÍSTUPŮ V TVORBĚ 3D MODELŮ	17
2.1.1 Digitální sochařství - 3D sculpting	18
2.1.2 Procedurální generování	18
2.1.3 Fotogrammetrie	19
2.1.4 3D skenování	20
2.2 POROVNÁNÍ PŘÍSTUPŮ V TVORBĚ TERÉNU	21
2.3 POROVNÁNÍ PŘÍSTUPŮ V TVORBĚ MATERIÁLŮ	22
2.3.1 Procedurální tvorba materiálů	22
2.3.2 Skenování materiálů	23
2.3.3 Kombinace skenovaných materiálů s procedurálními technikami	24
2.3.4 Ruční tvorba materiálů	24
3 PROCEDURÁLNÍ GENEROVÁNÍ V KOMBINACI S OSTATNÍMI TECHNIKAMI	25
4 NÁVRH A KONCEPCE PROCEDURÁLNÍCH SYSTÉMŮ	25
4.1 ANALÝZA	25
4.2 MODULARITA	26
5 NÁSTROJE PRO TVORBU PROCEDURÁLNÍCH SYSTÉMŮ	26
5.1 HOUDINI	26
5.2 BLENDER	27
5.3 SUBSTANCE DESIGNER.....	28
5.3.1 Netradiční užití Substance Designeru	28
II PRAKTICKÁ ČÁST	28

6	KRÁTKÝ FILM „POUŠŤ“	31
7	PROCEDURÁLNÍ TVORBA POUŠTNÍHO PROSTŘEDÍ	32
7.1	GENERÁTOR PÍSEČNÝCH DUN - HOUDINI	32
7.1.1	Rozbor generátoru písečných dun - Houdini	33
7.2	GENERÁTOR PÍSEČNÝCH DUN - SUBSTANCE DESIGNER	35
7.2.1	Rozbor generátoru písečných dun - Substance Designer	36
7.3	GENERÁTOR NEROVNOSTÍ PÍSEČNÉHO POVRCHU - SUBSTANCE DESIGNER	37
7.4	GENERÁTOR NÁHORNÍCH PLOŠIN - HOUDINI	38
7.4.1	Rozbor generátoru náhorních plošin - Houdini	38
7.5	ČÁSTICOVÉ EFEKTY	40
7.5.1	Problematika renderování kouře v Unreal engine	41
7.5.2	Zpracování simulace pro metodu Six Point Lighting	41
7.5.3	Zpracování simulace pro metodu 2D Raymarching	42
7.6	GENERÁTOR MRAKŮ - HOUDINI	43
8	PROCEDURÁLNÍ TVORBA PROSTŘEDÍ SCI-FI MĚSTA	45
8.1	GENERÁTOR PŮDORYSU SCI-FI MĚSTA - SUBSTANCE DESIGNER	46
8.2	GENERÁTOR SCI-FI MĚSTA - HOUDINI	46
8.3	GENERÁTOR SCI-FI BUDOV - HOUDINI	48
8.3.1	Rozbor generátoru sci-fi budov - Houdini	49
9	PROCEDURÁLNÍ TVORBA VESMÍRNÝCH LODÍ	50
9.1	GENERÁTOR VESMÍRNÝCH LODÍ - HOUDINI	51
10	PROCEDURÁLNÍ TVORBA MATERIÁLŮ	53
	ZÁVĚR	55
	SEZNAM POUŽITÉ LITERATURY	57
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	59
	SEZNAM OBRÁZKŮ	60

ÚVOD

Žijeme ve třetí dekádě 21. století, kdy filmový průmysl obklopuje každého z nás a denně ovlivňuje naši činnost. Drtivá většina populace každý den zapíná televizi či jiné audiovizuální zařízení a určitou dobu se věnuje sledování pořadů či filmů. Náročnost diváků stále roste a jejich očekávání a požadavky mají zpětně vliv na filmový průmysl. Ať si to uvědomujeme či nikoliv, v podstatě v každém filmu – bez ohledu na to, jestli to je pohádka, seriál ze současnosti, historický film, nebo science-fiction, se divák setká, i když si to mnohdy vůbec neuvědomuje, s vizuálními efekty, které dotvářejí charakter a děj filmové scény, nebo dokonce vytváří její základ.

Pokud se ohlédneme do minulosti, s praktikami vytváření filmového prostředí, které z finančních, časových a jiných důvodů nebylo možné kamerou natočit v reálné lokaci, se mohli pamětníci českých filmů setkat ve velké míře již v 60. letech minulého století. Vzpomeňme např. film *Baron Prášil*, který byl natočen režisérem Karlem Zemanem v roce 1962, či filmová zpracování románů Julese Vernea. Tehdy se samozřejmě ještě nepoužívala počítačová technika, ta přišla až o desítky let později.

S nástupem počítačů se otevřely nové možnosti tvorby prostředí. Některé techniky, jako například „dokreslovačky“ zůstaly v podstatě stejné - nekreslí se již na sklo umístěné před kamerou, ale digitálně v počítači. Jiné postupy však doznaly značných změn. V dnešní době lze díky výpočetnímu výkonu počítačů vytvářet naprosto fotorealistická prostředí kompletně ve 3D softwarech. Dobrým příkladem mohou být filmy jako *Avatar* (2009) Jamese Camerona, nebo nedávný remake *Lvího Krále* (2019) Jona Favreaux z tvorby Walt Disney Pictures. Pro zajímavost, film *Avatar* obsahoval přes 2500 záběrů, které využívaly vizuální efekty. Zmínovaný *Lví Král* je pro změnu vytvořen na počítačích již v plném rozsahu. A tento trend můžeme sledovat napříč filmovou tvorbou. Filmové efekty a počet záběrů, v nichž jsou využity, rok od roku nepřetržitě roste. Režiséři přicházejí se stále ambicióznějšími projekty, které mnohdy zobrazují rozsáhlá a komplexní virtuální prostředí jako například digitální města, přírodní scenérie, nebo dokonce celé světy.

Je pochopitelné, že takové množství záběrů využívajících vizuálních efektů vytváří extrémní nároky na postprodukční studia. Při současné poptávce po vysoce kvalitních a komplexních efektech se však umělci dostávají do situace, v níž nemají dostatek času na zdokonalení každé scény, což často může vést k vizuálním kompromisům, na které je nutno přistoupit, aby byly dodrženy termíny dohotovení filmového díla. . .

Právě zde přicházejí na pomoc procedurální techniky, jež dokáží značně urychlit tvorbu digitálních prostředí. Lze je použít od vytváření 3D modelů a jejich zasazování do prostředí, přes tvorbu materiálů (textur), až po tvorbu „chytrých“ generátorů, které umí např. obrůst 3D model domu břechťanem apod.

Princip procedurálních technik spočívá v tom, že se v softwaru vytvoří systém, který na základě vstupních parametrů a operacích s nimi generuje požadovaný výstup. Při návrhu systému je důležité najít kompromis mezi komplexitou a možnostmi generátoru, a to ideálně při zachování snadného ovládní. Tato práce ukazuje, jakým způsobem lze postupovat při návrhu systému tak, aby výsledný systém zajišťoval dobrou „uměleckou kontrolu“ nad výstupem, zároveň byl snadno modifikovatelný a rozšiřitelný. Teoretické poznatky jsou podloženy praktickými výstupy v podobě krátkého filmu, který ukazuje procedurálně generované pouště, města, vesmírné lodě a materiály jednotlivých povrchů. K tvorbě těchto systémů byly využity nástroje SideFX Houdini a Substance Designer. Práce rovněž demonstuje využití real-time technologií (Unreal Engine), které v současné době hýbou světem vizuálních efektů. Kombinace procedurálních technik s real-time technologiemi totiž skýtá obrovský potenciál a umožňuje velice efektivně produkovat komplexní prostředí v relativně krátkém čase.

I. TEORETICKÁ ČÁST

1 Procedurální přístup

S pojmem procedurální generování je možné setkat se v řadě odvětví. Tato práce má ale za cíl přiblížit problematiku právě v oblasti vizuálních efektů. Pojem procedurální přístup tak lze obecně definovat jako způsob či postup při vytváření objektů, prostředí nebo scén, které nejsou tvořeny ručně, ale jsou generovány dle nadefinovaných pravidel. Pod těmito pravidly je pak možno si představit sadu matematických operací a algoritmů, které generují výsledný produkt. Procedurální systémy také často využívají generátorů náhodných čísel, pomocí kterých je systém schopen generovat nekonečně mnoho variant v rámci stanovených pravidel generování.

Pro pochopení problematiky lze přístup ilustrovat na příkladu vytváření 3D modelu stromu. Pokud pomineme procedurální techniky, jednou z možností je ruční vymodelování s využitím například polygonálního způsobu modelování. Postupů je opět mnoho, ale hodně zjednodušený způsob by spočíval ve vytvoření válce, nebo kvádru, jenž by představoval kmen. Pokračovalo by se ručním natvarováním kmenu do požadovaného tvaru a vymodelováním hlavních větví, které z něho vyrůstají. Dále by se vymodelovaly jednotlivé menší větve, na něž by se poté ručně přidaly vymodelované listy.

Takto by mohl vypadat způsob, který nevyužívá žádné procedurální techniky. Je pravděpodobně zřejmé, že tento postup, ač by na jeho konci mohl být opravdu povedený model, není ideální, a to hned z několika důvodů. Prvním, a nejspíše hlavním problémem, by byla časová náročnost takového postupu. Vytvořit totiž jen samotné větve by totiž mohlo i zkušenému modeláři zabrat hodiny práce. Druhým problémem je fakt, že výsledkem je pouze jeden model konkrétního stromu.

Jen těžko si lze představit, kolik času by tímto způsobem zabralo vytvoření například celého lesa, kde je hned několik druhů stromů různých tvarů a velikostí. Přirozeně si tedy klademe otázku, zda by neexistoval lepší a elegantnější způsob tvorby stromů – systém, kde by stačilo zvolit druh stromu, jeho stáří, hustotu větví nebo typ podloží a systém by vygeneroval strom, který splňuje všechny tyto parametry. Odpovědí na tuto otázku jsou právě procedurální systémy.

1.1 Oblasti využití

Obecně se dá říct, že procedurální řešení nastupují tehdy, kdy se využití klasických ručních postupů a metod jeví jako neefektivní, nebo v některých případech dokonce prakticky ne-

možné. V oblasti vizuálních efektů, se procedurální techniky nejčastěji využívají právě k vytváření digitálních prostředí, což zahrnuje jak vytváření samotného terénu, tak 3D modelů objektů, jejich otexturování a zasazení do prostředí.

Příkladem využití síly procedurálních řešení může být například film *The Lion King* (2019, R: Jon Favreau), v němž bylo za pomoci procedurálních technik v kombinaci se satelitními snímky a fotogrammetrií vytvořeno 150 čtverečních kilometrů digitální, fotorealisticky vypadající africké krajiny. Procedurální techniky zde byly využity nejenom ke generování přírodních prvků, ale také k umístování jednotlivých objektů (stromy, rostliny, skály) do prostředí pomocí pravidel zvoleného ekosystému. Pokud tak umělec upravil nějakým způsobem terén, nebo změnil pozici některých objektů, oblast se sama přegenerovala, aby zohlednila změny a stále odpovídala nadefinovaným parametrům generování.¹

Procedurální řešení zde umožnily posunout i zaběhlé trendy v oblasti rozšíření scén pomocí digitálních „dokreslovaček“ (Matte Paint). V celém filmu je pouze okolo 10 záběrů, kde se Matte Paint využívá. Ve všech zbylých záběrech již bylo od této techniky upuštěno na úkor procedurálně generovaného 3D terénu. Výhodou tohoto postupu bylo, že tak kompozitoři získali nejen kompletní prostředí, ale také i jednotlivé render vrstvy, tzv. AOV's včetně přesných hloubkových dat (zDepth AOV), což ve výsledku usnadnilo proces kompozitingu.²

Ukázka prostředí využívající procedurální generování ve filmu *The Lion King* (2019) viz obrázek 1.1.

Typickým příkladem využití procedurálních technik mohou být také generátory měst - nejprve se vygenerují různé typy budov, které se následně umístí do vygenerovaného plánu, vytvoří se silnice, chodníky a celé město se zasadí do vygenerovaného prostředí. Ukázka procedurálně vygenerovaného města z filmu *Doctor Strange* (2016, R: Scott Derrickson) viz obrázek 1.2.

Dalším příkladem je využití procedurálních technik k vytváření tzv. „chytrých nástrojů“. Může se jednat například o generátory, které jsou schopny obrůst dům břečťanem, vytvořit oplocení okolo budov, nebo vygenerovat systém elektrických sloupů, vodovodních potrubí

¹BONATTI, Luca, Marco ROLANDI, Julien BOLBACH a Kai WOLTER. DMP without DMP, full-CG environments for *The Lion King*. In: ACM SIGGRAPH 2019 Talks [online]. ACM, 2019, s. 2. ISBN 9781450363174. Dostupné z: doi:10.1145/3306307.3328195

²BONATTI, Luca, Marco ROLANDI, Julien BOLBACH a Kai WOLTER. DMP without DMP, full-CG environments for *The Lion King*. In: ACM SIGGRAPH 2019 Talks [online]. ACM, 2019, s. 1. ISBN 9781450363174. Dostupné z: doi:10.1145/3306307.3328195



Obr. 1.1 Využití procedurálního generování ve filmu The Lion King (2019, R: Jon Favreau).



Obr. 1.2 Město vytvořené s pomocí procedurálních technik ve filmu Doctor Strange (2016, R: Scott Derrickson).

atd.

Procedurální přístup se rovněž využívá ke generování textur (tzv. materiálů), pro povrchy objektů. Využívá se zde základních geometrických tvarů, šumových funkcí, matematických operátorů a operací mezi nimi. Vznikají tak textury, které jsou vytvářeny kompletně v počítači a od těch reálných (fotografovaných) mohou být k nerozeznání.

1.2 Procedurální techniky v kombinaci s real-time technologiemi

Procedurální techniky přináší do oblasti vizuálních efektů efektivní způsob tvorby komplexních systémů. Pokud jsou tyto systémy správně navrženy, lze výstup generování snadno a rychle upravovat, a tak pružně reagovat na případné změny ve scénáři. Plné odemknutí potenciálu procedurálních technik však přichází teprve nyní, s příchodem real-time technologií do oboru vizuálních efektů.

V dnešní době se stále častěji setkáváme s pojmem virtuální produkce (virtual production). V zásadě se jedná o způsob produkce, která umožňuje nelineární, více iterativní proces tvorby. Svým způsobem lépe propojuje jednotlivé složky štábu se složkami VFX, což vede k přirozenějšímu procesu natáčení trikových záběrů.³ Pro ilustraci takového procesu lze zmínit např. seriál *The Mandalorian* (2019–2021, R: Jon Favreau), v němž místo tradičních zelených pláten využívají LED panely, na kterých se v reálném čase pomocí Unreal Engine renderují digitální prostředí. Výhoda takového procesu je asi jasná – nejen, že odpadá spousta práce v postprodukcí spojená s klíčováním zeleného pozadí, ale především herci dokáží adekvátně reagovat na prostředí, které díky LED panelům reálně vidí kolem sebe. Stejně tak kameraman může lépe komponovat záběr. Odpadá tedy nutnost si věci pouze představovat.

Další výhodou je, že digitální prostředí renderovaná na LED panely lze v reálném čase upravovat. A zde konečně přicházíme k obrovské výhodě procedurálních systémů – pokud můžeme „na place“ stiskem pouze jediného tlačítka generovat nekonečně mnoho variant prostředí, nebo úpravou několika parametrů měnit jeho charakter, získáváme obrovskou svobodu a volnost, která otevírá dveře kreativitě a vede k tvůrčím nápadům během samotného natáčení.

Do budoucna tak lze očekávat, že se díky virtuální produkci budeme ve filmech setkávat s mnohem přirozenější hereckou akcí, která bude zasazena v komplexních a propracovaných digitálních prostředích, jež budou díky pokrokům v real-time grafice budou k nerozeznání od těch reálných.

Kombinace procedurálních technik s real-time technologiemi je však přínosná i mimo virtuální produkci. Obecně lze díky real-time renderování mnohem rychleji iterovat a dojít proto k požadovanému výsledku dříve. Tato kombinace tak nyní otevírá dveře menším studiím,

³KADNER, Noah. *The Virtual Production Field Guide* [online]. Epic Games, 2019, s.7-8. Dostupné z: <https://www.unrealengine.com/en-US/vpfieldguide>

kteřá dokáží vytvářet komplexní digitální prostředí a za pomoci real-time engine je rendertovat v extrémně krátkém čase. Navíc k tomu nepotřebují ani obrovské render farmy. Díky tomu se i tato menší studia dokáží kvalitativně vyrovnat studiím s mnohonásobně větším rozpočtem a počtem lidí.

1.3 Výhody procedurálních systémů

Za největší výhodu procedurálního přístupu lze považovat možnost vytvářet komplexní systémy, které je možné snadno upravovat. Pod tímto si lze představit například tvorbu modelu, který lze kdykoliv – ať už během vývoje, nebo po dokončení – libovolně upravovat. Pokud je v některém kroku potřeba změnit nějaký parametr, neznamena to začít od daného místa znovu; změna se totiž automaticky vypropaguje skrz celou strukturu a promítne se do výsledného stavu. V případě procedurálního prostředí to může znamenat například možnost dynamicky měnit strukturu terénu, výskyt objektů apod.

Procedurální techniky mohou také značně zrychlit a usnadnit úkoly, manuálního charakteru. Jedná se právě o již zmiňované generátory, kterými lze například pokrýt objekty břechť a-nem, stanovit dva body, mezi kterými se mají vygenerovat potrubí, elektrické sloupy apod. Všechna tato řešení slouží v konečném důsledku k tomu, aby odbouraly repetitivní úkony, čímž umožní umělcům strávit více času nad kreativní stránkou věci.

Další nespornou výhodou je, že při korektním návrhu využívajícího myšlenku samostatných modulárních částí, lze snadno jednotlivé části opětovně použít v dalších projektech. Pokud například vytvoříme systém, který umí generovat převislé liány mezi stromy, lze ho lehce modifikovat ke generování drátů mezi sloupy elektrického vedení apod.

Jakožto drobnou výhodou lze považovat i většinou relativně malou velikost zdrojových souborů. Důvodem je fakt, že veškerá geometrie je (většinou) vygenerována dynamicky při otevření souboru. V případě 3D modelu není nutné ukládat například pozice jednotlivých vrcholů v prostoru. Podobně tomu je i v případě procedurálních textur. Právě textura s vysokým rozlišením může na disku zabírat i stovky megabytů. Výhodou procedurálních textur je tak podobně jako u 3D objektů fakt, že jsou v souboru definována pouze pravidla, podle kterých se má textura vygenerovat. Díky tomu lze celou knihovnu materiálů vměstnat do několika kilobytů, či megabytů, zatímco stejná knihovna uložená v rastrové podobě např. ve formátu .tga, či .png může na disku zabírat i stovky gigabytů.

1.4 Nevýhody procedurálních systémů

Použití procedurálních postupů se většinou nevyplatí v případě, kdy vytváříme pouze jeden konkrétní model. V případě procedurálního modelování totiž vytváříme předpis pro generování daného typu objektů. Díky tomu tak proces většinou trvá o něco déle, než kdybychom model vytvářeli ručně. Přidaná hodnota procedurálních systému spočívá v tom, že můžeme s jedním předpisem vygenerovat například sto odlišných verzí. Kdybychom tak potřebovali model jenom jeden, procedurální řešení nemá většinou své opodstatnění.

Dalším případem, ve kterém nelze použití procedurálních technik doporučit, jsou například situace, kdy chceme dosáhnout přesné reprezentace dle zadané předlohy. V takovém případě je potřebná plná kontrola nad výstupem, která procedurálními technikami nemusí být vždy snadno dosažitelná. Příkladem takové situace je právě „Pride Rock“ z filmu The Lion King (2019). Jedná se o ikonickou skalní formaci, kterou bylo potřeba vytvarovat tak, aby ze všech stran vypadala co nejlépe. Výsledný model byl vytvořen ručně v nástroji ZBrush.

2 Porovnání procedurálního přístupu s ostatními technikami

Způsobů, jak dosáhnout požadovaného výsledku je zpravidla hned několik. Každý přístup má však svá pro a proti, a tak je nutné důkladně zvážit, jakou cestu zvolit. Toto rozhodnutí nepadá většinou na samotné umělce, ale spíše na supervizory, kteří mají za úkol vytyčit nejvhodnější variantu v ohledu na možnosti týmu, nákladnost řešení a v neposlední řadě také časové možnosti. V tak komplexní oblasti, kterou z různých stran ovlivňuje řada faktorů, do které vizuální efekty bezesporu patří, mnohdy často vyhrává právě ta varianta, která je rychlejší, nebo umožňuje snadnou modifikaci pro případ opětovného využití v dalších projektech.

Pro srovnání přístupů a pochopení rozdílů mezi nimi, ilustrujme princip tvorby v několika příkladech, které spadají do tvorby digitálních virtuálních prostředí.

2.1 Porovnání přístupů v tvorbě 3D modelů

Jednoduchým příkladem pro pochopení rozdílů v přístupech modelování lze ilustrovat na principu vytváření 3D modelu skály.

2.1.1 Digitální sochařství - 3D sculpting

Ruční postup by pravděpodobně spočíval ve využití některého z nástrojů pro digitální sochařství (digital sculpting tools). Největší ikonou v této oblasti je profesionální nástroj ZBrush⁴ od vývojáře Pixologic. Proces spočívá v tvarování „digitální hlíny“, jejího přidávání a odřezávání. V podstatě je tento postup prakticky identický s reálným modelováním ze sochařské hlíny. Jedná se tak v zásadě o čistě umělecký přístup, který vyžaduje především dobrou znalost sochařských postupů a technik.

Na obrázku 2.1 lze vidět ručně vymodelovaný model skály (Pride Rock). Jedná se o známou skálu z filmu The Lion King (2019), která byla vytvořena pomocí nástroje ZBrush.



Obr. 2.1 Ukázka ručně vymodelovaného 3D modelu skály z filmu The Lion King (2019).

2.1.2 Procedurální generování

Naopak v případě řešení pomocí procedurálních nástrojů, je postup daleko techničtějšího rázu a vyžaduje spíše schopnost pochopit tvar a strukturu objektu a vztahy mezi jednotlivými částmi objektu. V tomto případě objekt nevzniká pod stylusem, kterým umělec ručně

⁴ZBrush je „bezkonkurenční jedničkou“ na poli nástrojů pro digitální sochařství. Díky své obrovské a neustále rozšiřující se sadě funkcí je využíván napříč VFX a herními studii po celém světě. <http://pixologic.com>

„vytesává“ jeho části, nýbrž vzniká kombinací matematických operátorů, využitím šumových funkcí k dodání nepravidelnosti tvarů apod. Pro vytvoření zmíněného modelu skály tímto způsobem by šlo využít například nástroj SideFX Houdini⁵, který je považován za favorita mezi nástroji pro procedurální tvorbu.

Na obrázku 2.2 lze vidět skálu, která byla vytvořena procedurálními technikami pomocí nástroje Houdini [3].



Obr. 2.2 Ukázka procedurálně generovaného 3D modelu skály.

2.1.3 Fotogrammetrie

Dalším způsobem, který se nabízí, je rekonstrukce reálné skály pomocí fotogrammetrie. Jedná se o poměrně náročný proces, který vyžaduje dobrou přípravu a znalosti z oblasti fotografování. Postup spočívá ve vytvoření snímků daného objektu s dostatečnými překryvy. V případě, že se objekt vyskytuje ve venkovním prostředí, a nemůžeme tak plně ovlivnit nasvícení objektu, je z důvodu eliminace přímých světelných zdrojů, které by zanesly do skenu nechtěné stíny, ideální fotografovat za zataženého počasí. Následně je fotografie nutné

⁵Houdini od vývojáře SideFX je, všestranný nástroj, který slouží k vytváření procedurálních systémů. V poslední době je možné pozorovat jeho expanzi do oblasti real-time herních enginů, což také napovídá o směřování odvětví počítačové grafiky obecně. Díky své robustnosti a flexibilitě je rovněž využíván napříč studii po celém světě. <https://www.sidefx.com/products/houdini/>

zpracovat v nástroji, který ve snímcích rozpozná stejné oblasti a provede 3D rekonstrukci. Mezi nejrozšířenější softwary patří Agisoft Metashape⁶, Reality Capture⁷ od slovenského vývojáře Capturing Reality, nebo free open-source nástroj AliceVision Meshroom.⁸ Vývoj v této oblasti jde ale rychle kupředu, a tak je pravděpodobné, že podobných softwarů bude do budoucna přibývat [4].

Po samotné rekonstrukci objektu z fotografií však stále proces nekončí. Model totiž většinou potřebuje mírně poupravit - zacelit díry v objektu, snížit počet polygonů a „zapéct“ informace o povrchu do textur. I když se jedná o relativně zdoluhavý proces, má jednu nespornou výhodu. Tímto postupem totiž automaticky získáme také materiál pro objekt, takže již není nutné vytvářet textury zvlášť. Pochopitelně i materiál vyžaduje jisté úpravy. Především difuzní (albedo) složka. I za ideálních podmínek je totiž ve snímcích vždy „zapečeno“ mírné ztmavení v zacloněných částech objektu. Nástroje, které model zpracovávají, však většinou dokáží pomocí analýzy geometrie objektu, nebo textur samotných i tyto nedokonalosti opravit. Výsledkem celého procesu fotogrammetrie je tak nejvěrnější reprezentace objektu, jaké lze v současné době dosáhnout. Největší rozdíl ve výsledném objektu oproti předešlým zmíněným metodám je viditelný především v detailech a různorodosti povrchu, který je sice předchozími metodami dosažitelný, ale časově extrémně náročný.

Na obrázku 2.3 lze vidět skálu, která byla vytvořena pomocí fotogrammetrie.

Jak je z obrázků 2.1, 2.2 a 2.3 patrné, ačkoliv se jedná o naprosto odlišné přístupy, využívající různé postupy a technologie, všemi těmito metodami lze dosáhnout detailního a realisticky vypadajícího modelu skály.

2.1.4 3D skenování

Nejpřesnější technologií pro přenos reálného objektu do digitální podoby je 3D skenování. Existují v zásadě dva způsoby skenování - skenování pomocí laseru (Laser 3D Scanning) a skenování pomocí projekce vzoru na objekt (Structured Light 3D Scanning). Tyto metody

⁶Agisoft Metashape je inteligentní nástroj pro zpracování fotogrammetrických dat.
<https://www.agisoft.com>

⁷Reality Capture nabízí přehledné a uživatelské rozhraní se spoustou pokročilých funkcí.
<https://www.capturingreality.com>

⁸AliceVision Meshroom je free open-source nástroj, který využívá k rekonstrukci objektu nodální systém. Tento přístup zajišťuje přehlednost a dobrou kontrolu nad výstupem.
<https://alicevision.org>



Obr. 2.3 Ukázka 3D modelu zrekonstruované skály pomocí fotogrammetrie.

se využívají především v případech, že je potřeba získat naprosto detailní objekt. Často se tak využívá právě v oblasti VFX, kde je nutné přenést reálný model do virtuálního prostředí v co nejvyšší kvalitě.

2.2 Porovnání přístupů v tvorbě terénů

Podobně jako 3D modely objektů i terény pro virtuální prostředí lze vytvářet jak ručně, tak také s využitím procedurálních technik.

Je pravdou, že v současné době se již od čistě ručního modelování terénů upouští. Nahrazují ho právě procedurální nebo semi-procedurální řešení, které celý proces značně urychlují. Nicméně pravděpodobně se stále budeme setkávat se situacemi, které budou vyžadovat plnou uměleckou kontrolu nad výsledkem. V těchto ohledech tak mohou být nástroje využívající procedurální generování za jistých okolností limitující.

Základní rozdíl mezi ruční a procedurální tvorbou terénu je ten, že u procedurálních řešení nevytváříme terén na přímo, ale spíše definujeme obecnou strukturu, kterou by měl terén mít. Nevýhodou tak může být, že s plně procedurálním řešením bývá často poměrně obtížné dosáhnout přesně požadovaného výsledku. Procedurální generování terénů totiž spočívá na využití šumových funkcí, které sice nabízejí nekonečně mnoho variant v závislosti na tzv. semínku pro generování (seed), nicméně díky náhodnému charakteru šumu, nelze přirozeně dosáhnout přesně požadované představy.

Zde tak přicházejí tzv. Directed Procedural Workflows, které kombinují procedurální přístup s možností ručního zásahu do generování. V praxi to pak může znamenat, že lze použít ště-

tec, kterým je možno vyznačit oblasti např. pro hory jenž jsou posléze dotvořeny již pomocí procedurálních technik. Lze tak zajistit, že rozvržení může přesně odpovídat požadované předloze, zatímco samotné hory jsou již generovány automaticky, bez nutnosti dalších ručních zásahů [5].

V případě, že je terén hotový a je potřeba do něj zasadit další objekty, je opět možné využít plně ručního čistě procedurálního řešení, nebo kombinací obou předchozích metod. Poslední zmíněný postup je většinou opět jakýmsi zlatým středem, poněvadž si bere z obou technik to nejlepší – zajišťuje dostatečnou „uměleckou kontrolu“ nad výstupem při zachování výhod procedurálních systémů.

2.3 Porovnání přístupů v tvorbě materiálů

Zatímco dříve se umělci museli spoléhat na tvorbu textur v grafických editorech à la Photoshop, v současné době existuje mnoho nástrojů, které jsou pro tvorbu textur, potažmo materiálů přímo určeny. Při tvorbě materiálů se můžeme setkat v podstatě se čtyřmi základními principy.

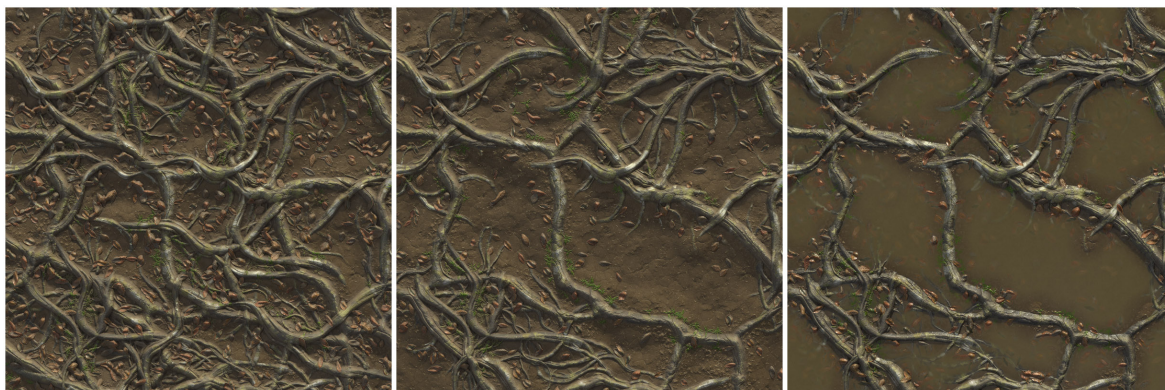
2.3.1 Procedurální tvorba materiálů

I když se procedurální generování textur využívá již celá desetiletí, největší rozmach znamenalo až s nástupem softwaru Substance Designer,⁹ ze kterého se stal tzv. „industry standard“.

Procedurální materiály mají řadu výhod. Při tvorbě můžeme pomocí parametrů materiály libovolně modifikovat a přizpůsobovat je konkrétnímu prostředí. Situaci ilustruje obrázek 2.4, kde lze pomocí posuvníku dynamicky měnit množství kořenů, hustotu napadaného listí a nebo výšku vodní hladiny. Tvůrce materiálu může nastavit a zpřístupnit libovolný počet takovýchto parametrů. Možnosti jsou prakticky neomezené.

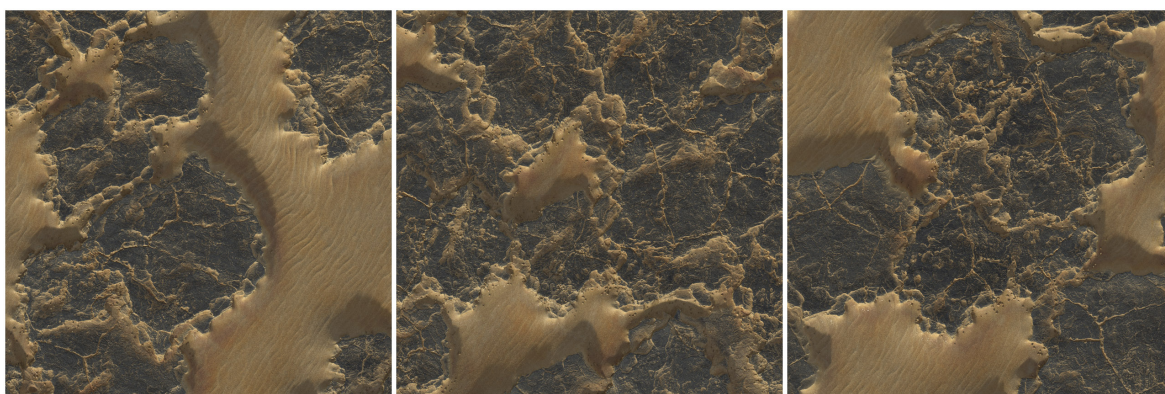
Další výhodou je automatické zajištění návaznosti textury při jejím opakování na velké ploše (seamless tiling). Díky tomu, že jsou materiály tvořeny šumovými funkcemi, matematicky popsány tvary a operátory, lze textury rovněž generovat v teoreticky libovolném rozli-

⁹Substance Designer je nástroj sloužící k vytváření parametrických procedurálních materiálů. Lze ho ale také využít ke zpracování skenovaných materiálů, či tvorbě výškových map pro terény. Jedná se o velice flexibilní nástroj, který je nepostradatelnou součástí pipeline v mnoha studiích. <https://www.substance3d.com>



Obr. 2.4 Ukázka různých hodnot parametru procedurálně generovaného materiálu.

šení¹⁰. Pochopitelně jsou zde jisté hardwarové limity, například v podobně omezené velikosti paměti, nicméně i se současným hardwarem lze textury vytvářet i v 8K nebo 16K rozlišení. Materiály lze také jednoduše variovat díky změně tzv. seedu v generátorech. Prakticky to znamená, že lze vytvořit nekonečně mnoho materiálů stejného typu, které jsou ale vždy trochu odlišné. Situaci ilustruje obrázek 2.5, na němž je vyobrazen jeden materiál ale s různými „seedy“.



Obr. 2.5 Ukázka různých verzí procedurálního materiálu (změna hodnoty „random seed“).

2.3.2 Skenování materiálů

Další možností tvorby materiálu je skenování. Skenované materiály se začaly rozšiřovat především v návaznosti na pokroky v oblasti tzv. Physically Based materiálů, u kterých je snahou co nejpřesněji přenesení měřitelných fyzikálních parametrů reálných povrchů do digitální podoby. Opět se jedná o poměrně složitý proces vyžadující správné vybavení a dobré

¹⁰OKUN, Jeffrey A. a Zwerman, Susan. The VES Handbook of Visual Effects: Industry Standard VFX Practices and Procedures. Focal Press, 2010. s. 617. ISBN 978-0-240-81242-7

znalosti problematiky.

Existuje několik způsobů, jak reálný materiál oskenovat. Prvním způsobem je využití tzv. PBR Texture Scanneru. Skenování funguje na principu, že je povrch vloženého objektu nasvícen z různých stran a jednotlivě vyfotografován. Vznikne tak několik fotografií (většinou 4, nebo 8, v závislosti na pokročilosti skeneru) s různým úhlem osvětlení, ze kterého je následně možno v dedikovaných softwarech rekonstruovat strukturu povrchu.

Pro povrchy, které jsou již více členité je vhodnější použít například laserové 3D skenování, nebo fotogrammetrii. Podobně jako tomu je u skenovaných objektů, materiály získané pomocí 3D skenování, nebo fotogrammetrie, jsou nejpřesnější reprezentací materiálů z reálného světa. Postup je do značné míry podobný jako v případě rekonstrukce 3D modelů a tudíž není nutné ho zde znovu rozebírat - viz podkapitoly 2.1.3 a 2.1.4.

2.3.3 Kombinace skenovaných materiálů s procedurálními technikami

Zajímavou variací na skenované materiály přináší nástroj Quixel Mixer¹¹. Jedná se o velmi umělecky přívětivý způsob tvorby materiálů, který si vypůjčuje některé procedurální a ruční techniky a mixuje je se skenovanými materiály.

Princip funguje na kombinaci několika materiálů pomocí ručních nebo automatických „chytrých masek“, čímž vznikají zcela nové materiály. Lze tak například smíchat materiál skály s materiálem písku, který lze pomocí chytré masky aplikovat pouze do štěrbin a záhybů, takže vznikne nový materiál, který připomíná skalní podloží s nánosy písku. Kombinací různých materiálů, masek a filtrů je možné vytvářet prakticky nekonečně mnoho variant materiálu.

2.3.4 Ruční tvorba materiálů

Ruční tvorbou se v tomto případě chápe vytváření materiálů buď v programech typu Photoshop, nebo například ručním vymodelováním (např. v nástroji ZBrush) a následném „zapečení“ do textur. S nástupem procedurálních technik ale tyto postupy pomalu upadají. Využívají se ale stále hojně v oblasti stylizovaných či kreslených materiálů.

¹¹Quixel Mixer je po spojení se s Epic Games volně dostupný nástroj k vytváření materiálů formou míchání existujících materiálů za pomoci ručně či procedurálně vytvořených masek. Kromě tvorby materiálů lze nástroj využít i k texturování objektů. <https://quixel.com/mixer>

3 Procedurální generování v kombinaci s ostatními technikami

Pochopitelně, že v praxi často dochází ke kombinaci výše zmíněných technik. Ideální je využít z každého přístupu to nejlepší a vhodně tyto dílčí výstupy zkombinovat. Např. vytvořit základní tvar skály ručně (ZBrush) a následně využít procedurálních technik k doplnění malých povrchových nerovností. Tímto způsobem lze definovat přesný tvar skály a následně využít procedurálních technik tam, kde je to pracné a divák to povětšinou stejně nepozná (např. právě jemná struktura na skále).¹²

V případě generování prostředí se nabízí možnost kombinovat procedurální techniky s reálnými daty. Využít například reálné výškové mapy terénu a do nich procedurálně umístit objekty.

Při vytváření městských prostředí se vybízí využít reálných mapových podkladů ulic. Data je možno získat například z projektu OpenStreetMap.¹³ Jelikož se koncepčně jedná o tzv. Open Source je možné data volně získat pod licencí „Open Database Licence“ (ODbL). Ke stažení topografických údajů tak stačí vyznačit danou oblast a mapu vyexportovat. Data lze získat v různých formátech, přičemž nejsnazší je asi práce s formátem XML, který lze relativně snadno parsovat.

4 Návrh a koncepce procedurálních systémů

Návrh je prvním a zároveň jedním z nejdůležitějších kroků při tvorbě procedurálních systémů. Právě korektním návrhem je možno využít všechny výše zmíněné výhody, které procedurální systémy nabízí. Následující odstavce tak odpovídají na otázku, jakým způsobem efektivně navrhovat procedurální systémy, které jsou zároveň přehledné a snadno modifikovatelné.

4.1 Analýza

Důležitým krokem při návrhu jakéhokoliv procedurálního systému je rozpoznání a definování dílčích částí, který dávají dohromady celek. Tento krok je velice důležitý, protože umož-

¹²OKUN, Jeffrey A. a Zwerman, Susan. The VES Handbook of Visual Effects: Industry Standard VFX Practices and Procedures. Focal Press, 2010. s. 678. ISBN 978-0-240-81242-7

¹³OpenStreetMap je projekt, který nabízí volně dostupné mapové podklady. Mapy může kdokoliv upravovat a jelikož jsou distribuovány pod licencí ODbL, je možné je volně využívat bez poplatků. www.openstreetmap.org/

ňuje rozebrat složitý problém do několika dílčích částí, které se již řeší daleko snáz. Určením a případným rozkreslením těchto jednotlivých částí si ujasníme i jednotlivé souvislosti mezi komponenty, a lépe tak dokážeme předvídat případné problémy, které mohou při tvorbě systému nastat.

4.2 Modularita

Jednou z hlavních bodů při návrhu by měla být modularita. Základní myšlenkou je, že výsledný systém by měl být složen ze samostatně nezávislých částí, které ve finále tvoří celek. Výhoda takového přístupu poté spočívá v tom, že lze jednotlivé části nezávisle na sobě kombinovat a opětovně využívat v dalších systémech. Další nespornou výhodou je snazší orientace, především u komplexnějších systémů. Pokud je celek rozložen do několika logicky oddělených částí, snáze se s takovým systémem pracuje a usnadňuje to také hledání případných chyb a problémů.

V reálné praxi to poté znamená např. vytvoření systému, který generuje základní tvar objektu, filtr, který základnímu tvaru přidá detaily a následně systém, který objekt vezme a umístí ho do scény. Pokud tyto jednotlivé nezávislé části spojíme v jeden celek vznikne nám např. generátor, skal do virtuálního prostředí. Díky modulárnímu návrhu je poté možné vyměnit některou část za jinou a tímto způsobem kombinovat a vytvářet nové systémy.

5 Nástroje pro tvorbu procedurálních systémů

Tato kapitola se věnuje několika nástrojům pro procedurální generování. Nejedná se o kompletní výčet všech softwarů, ale o základní představení těch nejrozšířenějších, které byly zároveň využity v praktické části práce.

5.1 Houdini

Houdini je nejrozšířenějším nástrojem v oblasti procedurálně generovaných systémů a efektů obecně. Využívá se také často pro různé simulace (voda, oheň, kouř apod.). K tvorbě procedurálních systémů využívá tzv. nodální systém.¹⁴ Jedná se o způsob, se kterým se v softwarcích pro práci s 3D grafikou setkáváme stále častěji. Je to především z toho důvodu, že tento systém zajišťuje dobrou přehlednost a orientace v něm je tak snadná. Jednotlivé

¹⁴OKUN, Jeffrey A. a Zwerman, Susan. The VES Handbook of Visual Effects: Industry Standard VFX Practices and Procedures. Focal Press, 2010. s. 699. ISBN 978-0-240-81242-7

operátory si lze představit jako „krabičky“, které jsou pomocí spojení připomínající „dráty“ pospojovány ve výslednou síť, potažmo graf. Velkou výhodou nástroje Houdini je i přímá integrace vlastního skriptovacího jazyku VEX. Svou syntaxí připomíná jazyk C/C++, a tak je snadno uchopitelný pro většinu programátorů nebo lidí, kteří se s programováním již setkali. Díky této integraci nabízí uživateli možnost psaní vlastních skriptů jakožto samotnou součást grafu. Pomocí VEX kódu lze v Houdini pracovat jak s geometrií 3D objektů (SOPs), částicovými systémy (POPs), při práci s renderovacím enginem mantra a následnou kompozicí (COPs), při zpracovávání časových dat např. v animacích (CHOPs) nebo například pro manipulaci a stylizaci digitálních vlasů a srsti (Hair and Fur).

Obrovské možnosti, které Houdini uživateli nabízí jsou však promítnuty do strmější křivky učení. Vyzdvihnout zde však lze kvalitně zpracovanou dokumentaci a také vstřícnou komunitu uživatelů, kteří at' už na oficiálních nebo neoficiálních fórech ochotně sdílí své znalosti s ostatními uživateli. Dobrým základem pro seznámení se s Houdini je také publikace Houdini Foundations, která rovněž obsahuje několik ukázkových lekcí, z nichž jedna je věnována i procedurální tvorbě assetů pro Unreal Engine.¹⁵

5.2 Blender

Ačkoliv se v době vydání této práce nejedná o software, který by byl automaticky spojován s procedurálními řešeními, je velice pravděpodobné, že se jím do budoucna stane. Od verze 2.92, která vyšla na konci roku 2020 přichází s tzv. „geometry nodes“. Jedná se o operátory, které svou funkcionalitou a způsobem implementace hodně připomínají nástroj Houdini. Zjednodušeně by se dalo říct, že se jedná o volnou vstupenku do světa procedurálních řešení. Blender totiž i nadále zůstává Free Open Source softwarem.

Podobně jako tomu je v nástroji Houdini, využívá Blender nodální systém, což se pro tyto případy obecně ukazuje jako ideální řešení. Ostatně nástroje jakými jsou právě Houdini, Substance Designer, nebo z oblasti kompozitorů Nuke nebo Fusion to jenom potvrzují. Pomocí propojování atomických operátorů tak vzniká komplexní síť, která může manipulovat s objekty ve scéně. Podobně jako například v Houdini tak lze nyní i v Blenderu vytvářet různé generátory budov, zástaveb, stromů, distribuovat objekty po ploše apod.

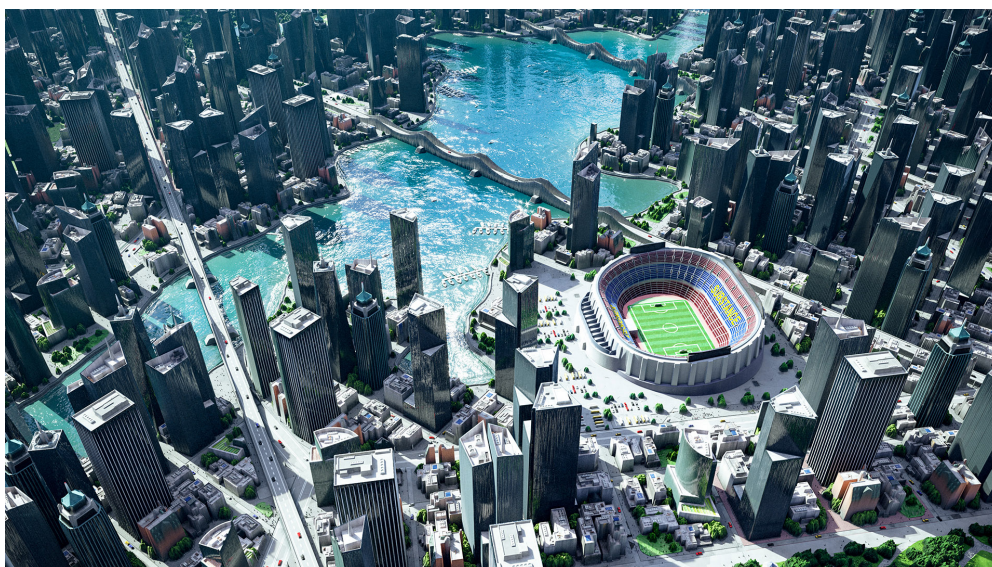
¹⁵MAGEE, Robert, Houdini Foundations for Film, TV & Gamedev. SideFX Software, 2018. s. 69-84. ISBN 978-1-7753338-0-7. Dostupné také z: https://www.sidefx.com/media/uploads/tutorial/foundations_gdc2018/houdini_foundations.pdf

5.3 Substance Designer

Substance Designer lze považovat za jeden z nejrozšířenějších nástrojů v oblasti procedurálně generovaných materiálů. Díky své všestrannosti je hojně využíván napříč filmovými i herními studii po celém světě. Podobně jako Houdini i Substance Designer sází na nodální systém. Pomocí jednotlivých operátorů a šumových funkcí tak lze generovat prakticky libovolné tvary a struktury. Jejich kombinací lze pak vytvářet materiály povrchů, které mohou být prakticky k nerozeznání od těch skenovaných. Na rozdíl od skenů je ale lze libovolně upravovat a modifikovat (viz kapitola 2.3.1). Další výhodou je možnost přímé manipulace s jednotlivými pixely obrazu za pomoci operátoru *Pixel Processor*. Se znalostmi z oblasti digitálního zpracování obrazu, tak lze skriptovat obrazové operace jako jsou například konvoluce, vytvářet hranové detektory za pomoci derivací jasových funkcí obrazu apod.

5.3.1 Netradiční užití Substance Designeru

Obrovskou výhodou Substance Designeru je jeho univerzálnost. Díky ní ho lze použít i k prototypování. Všestrannost tohoto nástroje ilustrují následující obrázky 5.1, 5.2 a 5.3. Kompletní prostředí na obrázcích je vytvořeno v Substance Designeru. Žádné další 3D modely nebyly použity.



Obr. 5.1 Generátor metropolitních měst vytvořen v nástroji Substance Designer.



Obr. 5.2 Generátor středověkého města vytvořen v nástroji Substance Designer.



Obr. 5.3 Generátor terénu pokrytého lesy vytvořen v nástroji Substance Designer.

II. PRAKTICKÁ ČÁST

6 Krátký film „Poušť“

Diplomový projekt s pracovním názvem „Poušť“ demonstruje využití procedurálních technik ke generování virtuálních prostředí pro oblast filmového triku. Jedná se o krátký snímek, který zobrazuje nespoutanost, drsnost a zároveň estetickou krásu pouštní krajiny. Kontrastním elementem je prostředí industriálního sci-fi města, které se nachází uprostřed pouště.

Hlavním cílem snímku je poukázat na možnosti a potenciál těchto technologií, jež umožňují i malému týmu lidí, či jednotlivci vytvářet komplexní prostředí, která jsou při správném použití navíc snadno modifikovatelná, a zajišťují tak flexibilitu, která je v procesu tvorby vizuálních efektů velice důležitá.

Současný trend vývoje real-time grafiky je velmi strmý, a tak lze předpokládat, že se v následujících letech stane tato technologie standardem i pro oblast vizuálních efektů. Z tohoto důvodu je výstup praktické části práce zaměřen právě na tyto technologie. Výsledné audiovizuální dílo je renderováno v reálném čase za pomoci Unreal Engine 5, který vyšel v průběhu psaní této práce. Díky zpětné kompatibilitě s Unreal Engine 4, bylo možné projekt konvertovat pro novou verzi, a využít tak všechny novinky, které v pořadí již pátá verze enginu přináší.

Kompletní prostředí pro film, včetně jednotlivých budov, vesmírných lodí, některých částicových efektů, textur a materiálů, bylo vytvořeno v nástrojích Houdini a Substance Designer. Postupy a techniky, které byly při procesu využity jsou popsány v následujících kapitolách.

Jednu z výsledných scén filmu přibližuje obrázek 6.1



Obr. 6.1 Výsledné prostředí industriálního města z filmu vytvořeno za pomoci procedurálních technik. Renderováno v reálném čase v Unreal Engine 5.

7 Procedurální tvorba pouštního prostředí

Referencí pro tvorbu prostředí byly satelitní snímky a fotografické materiály z Afriky, konkrétně z oblasti národního parku Tassili n'Ajjer, v Alžírsku. Snahou bylo zachytit různorodost písečných formací, které nabývají různých tvarů a velikostí, určených místem výskytu, silou a směrem větrného proudění. Ukázka části obrazové předlohy, která byla využita pro tvorbu generátorů viz obrázek 7.1.



Obr. 7.1 Ukázka části obrazové předlohy použité k tvorbě generátorů.

Duny byly pro film generovány dvěma způsoby, a to dle konkrétního typu záběru. První způsob spočíval ve využití nástroje Houdini, ve kterém byla generována prostředí dun o velké rozloze – řádově v desítkách kilometrů čtverečních. Druhý způsob využíval Substance Designer a sloužil primárně ke generování menších a detailnějších ploch o rozloze v jednotkách kilometrů čtverečních, či úplně samostatných dun.

Ukázka výsledného prostředí pouště po importu v Unreal Engine 5 viz obrázek 7.2. Pro prostředí vzniklo kombinací generátorů vytvořených v Houdini a Substance Designeru.

7.1 Generátor písečných dun - Houdini

Houdini disponuje poměrně robustním systémem pro generování terénu – *Houdini Height-Fields*. Pomocí tohoto systému bylo vytvořeno hned několik prostředí pro různé části filmu. Přestože, se v reálném světě můžeme setkat s písečnými formacemi, kde se na velké ploše mnohokrát opakuje tentýž vzor, pro účely filmu bylo snahou se takovým formacím spíše vyhnout a to především z toho důvodu, že pro diváka mohou působit uměle a nezajímavě.



Obr. 7.2 Pouštní duny vytvořené procedurálními technikami v Houdini a Substance Designeru. Vyrenderováno v reálném čase v Unreal Engine 5.

Snahou tedy bylo vytvořit co nejrozmanitější formace, které jsou typické pro oblasti, kde se častěji mění směr proudění větru a jsou tak pro diváka zajímavější.

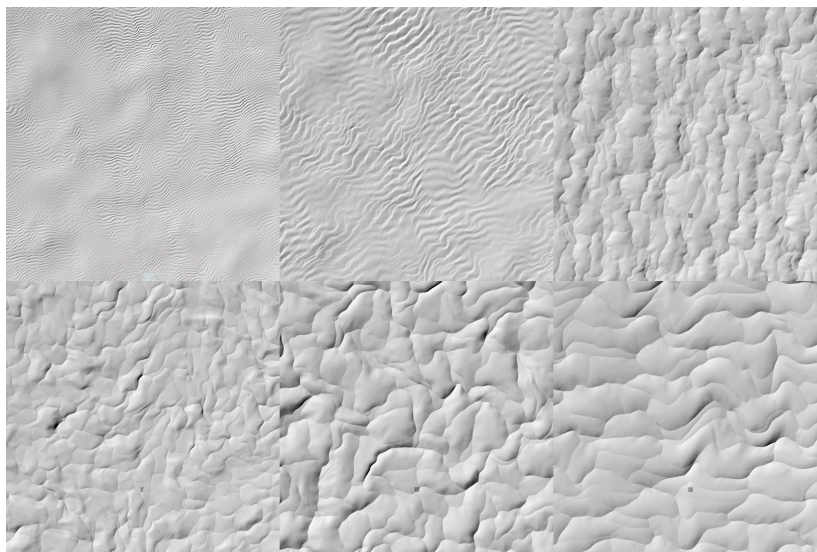
7.1.1 Rozbor generátoru písečných dun - Houdini

Výsledné prostředí je složeno z šesti na sobě nezávislých sub-generátorů, které reprezentují různé podoby písečných formací. Díky tomu je možné je libovolně upravovat nebo zaměňovat a případná změna se projeví napříč celým prostředím, bez nutnosti dalších zásahů. Tyto základní sub-generátory jsou v zásadě poměrně jednoduché. Základními prvky jsou operátory *HeightField Noise*, nejčastěji využívající šumovou funkci typu *Worley (cellular) F1*. Tento základní tvar je následně pomocí několika operátorů *HeightField Distort* zohýbaný a různě zvrásněný, tak aby výstup nepůsobil jednotvárně.

Ukázka výstupů jednotlivých sub-generátorů viz obrázek 7.3. Obrázek znázorňuje pouze reliéf a tudíž má šedou barvu. Výsledný materiál byl na povrch aplikován až po importu do Unreal Engineu.

Následně byly tyto jednotlivé generátory použity jako vstupy do hlavního generátoru, který mezi sebou tyto vzory prolínal a kombinoval dle jejich charakteru do výsledného celku.

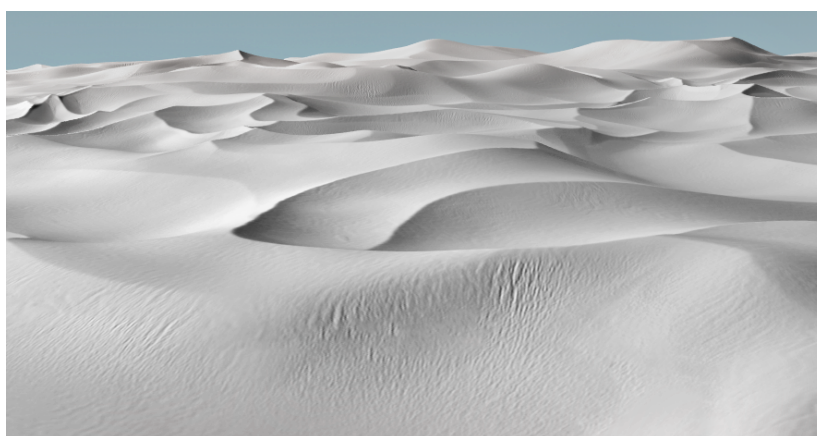
Pro zajištění dostatečné kontroly nad generováním, které by umožnilo přizpůsobit prostředí požadavkům záběru, je velikost dun určena maskou, kterou lze nakreslit přímo v Houdini s využitím operátoru *HeightField Draw Mask*. Plynulé prolnutí jednotlivých vrstev je pak zajištěno pomocí detekce a izolace okrajových částí masky, které pak slouží jako místo, kde má dojít k prolnutí. V dalším kroku je pomocí operátoru *HeightField Mask Expand*



Obr. 7.3 Ukázka jednotlivých sub-generátorů písčných formací, vytvořených v nástroji Houdini.

maska extrapolována a proces se opakuje pro další vrstvu (sub-generátor). Tímto způsobem je v automaticky generovaném procesu zachována možnost autorského zákroku.

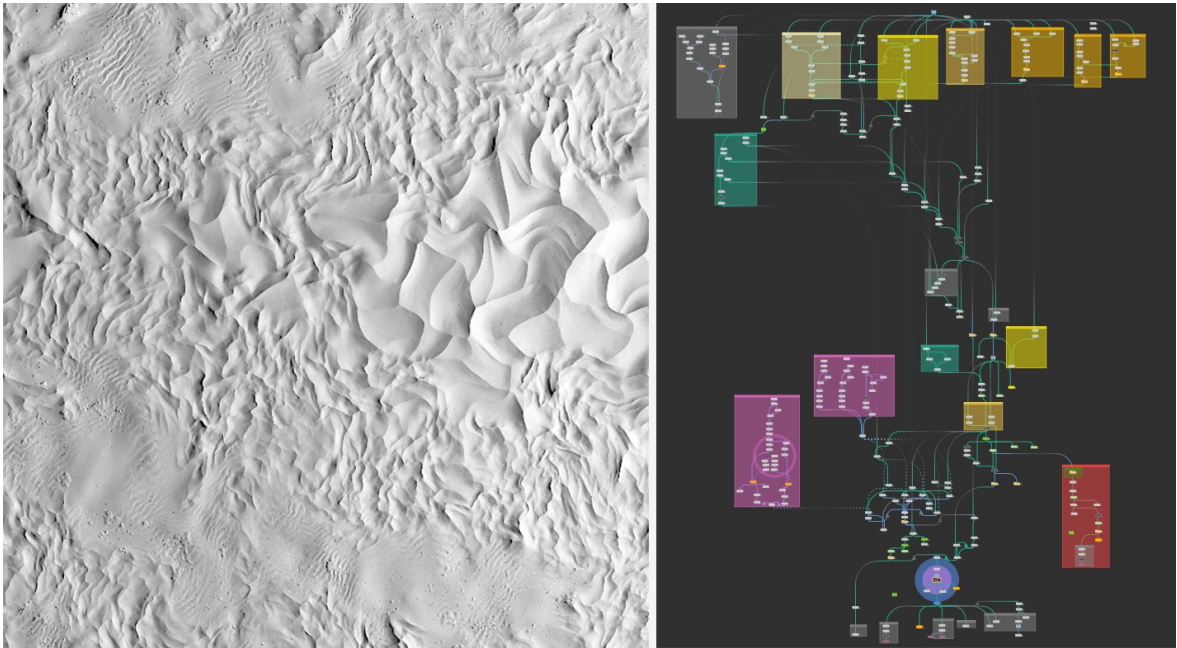
Po fázi, kde došlo k prolnutí jednotlivých sub-generátorů, bylo aplikováno několik šumových funkcí, které pomohly výsledný povrch ještě lépe zcelit. Pro dosažení jemných detailů na povrchu byl na závěr ještě terén několikrát erodován pomocí operátorů *HeightField Erode*. Výsledný efekt jemného zvrásnění v závětrných částech dun je patrný na obrázku 7.4



Obr. 7.4 Ukázka jemného zvrásnění dosaženého pomocí simulace eroze v nástroji Houdini.

Jedna z výsledných variant celkového finálního prostředí o rozloze 25km² včetně grafu generátoru je znázorněna na obrázku 7.5.

Jednou z komplikací při práci s velkými a detailními terény je náročnost na velikost operační



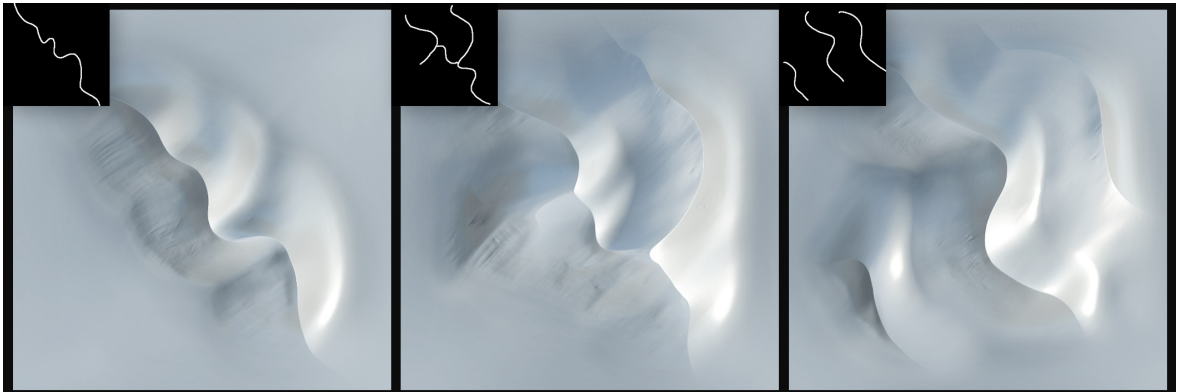
Obr. 7.5 Ukázka výsledného povrchu pouště o rozloze 25km² vytvořeného kombinací jednotlivých sub-generátorů a jeho výsledný graf v nástroji Houdini.

paměti počítače (RAM). Houdini v tomto směru ale umožňuje uložení jednotlivých částí na pevný disk formou cache. Znamená to, že pokud víme, že danou část grafu již nebudeme nijak modifikovat, můžeme ji uložit na disk a při další práci již číst tato data z disku. Tím odpadne nutnost danou část znovu počítat, čímž dojde i k úspoře operační paměti.

7.2 Generátor písečných dun - Substance Designer

I když je nástroj Substance Designer určen primárně ke generování materiálů, díky jeho všestrannosti ho lze využít i ke generování výškových map pro terén. Byl tak vytvořen generátor, který dle požadovaného tvaru vytvoří realisticky vypadající písečnou dunu. Vstupem generátoru je jednoduchý obrázek, který schématicky znázorňuje tvar písečné duny. Obrázek lze vytvořit v libovolném programu. Generátor poté tvar zanalyzuje a vytvoří výškovou mapu duny. Tímto způsobem byly vytvořeny jak jednotlivé duny, tak také větší písečné formace. Pomocí parametrů, jako je např. směr a intenzita větru, intenzita zvrásnění atd. lze měnit výsledný vzhled duny. Ukázku vstupů a výstupů generátoru znázorňuje obrázek 7.6.

Vytvořené výškové mapy byly poté naimportovány do Unreal Engine a s využitím nástroje *Landscape Tool* následně „razítkovány“ po ploše. Tato možnost namapování libovolné textury (výškové mapy) na štětec a jejího razítkování po terénu se ukázala jako ideální kombinace procedurálního a ručního řešení. Díky generátoru bylo vytvoření libovolného tvaru



Obr. 7.6 Ukázka výsledných dun generátoru, které jsou generovány na základě jednoduchých vstupů (jednoduchá schémata v rozích) - Substance Designer.

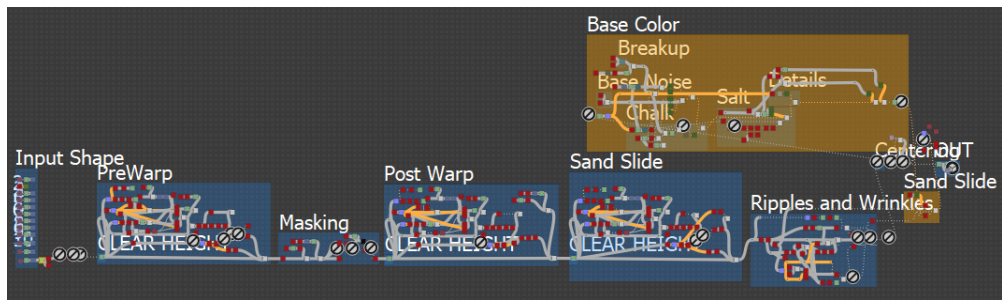
duny snadné a pomocí ručního razítkování ji bylo možné umístit do scény přesně podle potřeby záběru. Na tomto příkladu lze tedy opět vidět, že „procedurální“ neznamená nutně „náhodně generované“. Vhodnou kombinací lze využít rychlost procedurálního řešení, ale mít stále plnou kontrolu nad výstupem.

7.2.1 Rozbor generátoru písečných dun - Substance Designer

Nejprve byla vstupní tenká čára představující tvar duny rozšířena do požadované šířky pomocí operátoru *Distance*. Následně byla provedena první iterace eroze, která dala duně základní charakteristický tvar. Tento efekt byl dosažen za pomoci vlastního generátoru eroze, který dle zvoleného směru větru simuluje efekt eroze na povrchu. V tuto chvíli měla duna již patřičný tvar, nicméně pro dosažení realističtějšího vzhledu bylo aplikováno ještě několik filtrů které především za pomoci *Vector Warp* a *Non-Uniform Dir. Warp Grayscale* operátorů napomohly narušit příliš pravidelný tvar. Na závěr byl na závětrné části duny vytvořen efekt rozbrázděného písku vlivem písečného sesuvu. Klíčové tak bylo vymaskovat patřičnou stranu duny s dostatečně strmým sklonem a v těchto místech vytvořit požadovaný efekt. Způsobů, jak vytvořit takovou masku je několik, nicméně nejefektivnější způsob spočívá ve využití operátoru *Color to Mask* na jehož vstupu je normálová mapa. Následně lze kliknutím na vybrané místo na normálové mapě vybrat patřičné místo pomocí kapátka. Dále je potřeba nastavit dostatečný rozsah a případně pomocí operátoru *Blur HQ Grayscale* masku vyhladit. Poté již stačí do vymaskované oblasti prolnout texturu připomínající písečný sesuv. V tomto případě byla tato textura vytvořena opět pomocí vlastního generátoru eroze.

Při tvorbě komplexnějších generátorů, jakým je například tento, je vhodné dbát na modularitu jednotlivých částí. Pro lepší přehlednost byly tyto části ještě zarámovány do vlastních

rámečků, což je patrné na obrázku 7.7.



Obr. 7.7 Graf generující pouštní duny v Substance Designeru.

7.3 Generátor nerovností písčného povrchu - Substance Designer

Pro potřebu filmu bylo vytvořeno i několik textur, které simulují různé povrchové nerovnosti na dunách nebo okolo různých objektů (např. trsů trávy). Přestože, byl původní terén v poměrně vysokém rozlišení, drobných povrchových nerovností díky hardwarovým omezením stále nebylo možné dosáhnout. Řešením tak byly tyto textury, které byly formou *Deferred Decal Actor* rozmístěny do scény a promítnuty na terén. Ukázka několika takových textur viz obrázek 7.8. K těmto texturám byly rovněž vygenerovány odpovídající normálové mapy, díky kterým simulované povrchové nerovnosti adekvátně reagovaly na světelný zdroj ve scéně.



Obr. 7.8 Ukázka vygenerovaných textur různých typů povrchových nerovností - Substance Designer.

Ukázka srovnávající stav před a po aplikaci těchto textur znázorňuje obrázek 7.9. Zde je pomocí těchto textur simulován nahromaděný písek za trsy trávy. Jedná se o drobný detail, nicméně naakumulovaný písek za trávou ve směru větru dodává na realističnosti. Bez tohoto efektu působí tráva umělým dojmem a působí spíše jako kdyby někdo trsy právě rozházel po zemi.

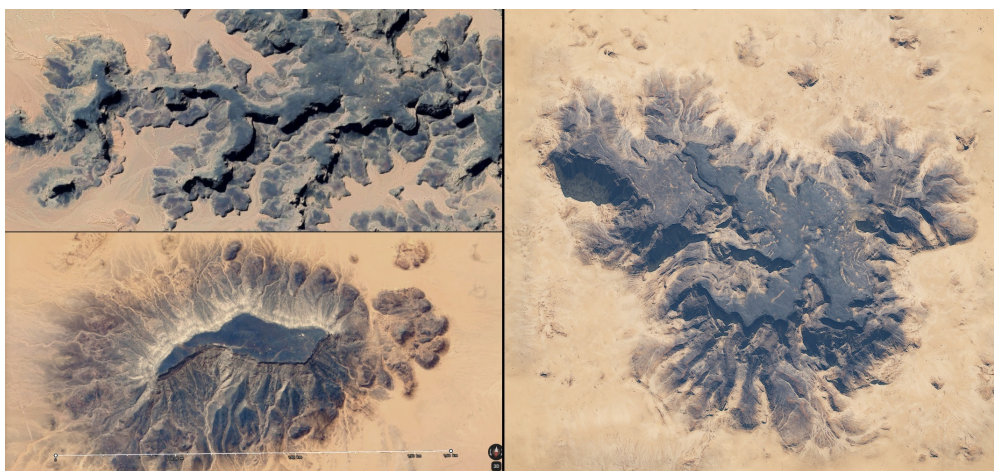


Obr. 7.9 Srovnání scény před a po aplikaci textur simulující drobné povrchové nerovnosti.

7.4 Generátor náhorních plošin - Houdini

Pod pojmem pouštním prostředí si však nelze představit pouze písečné planiny. Především pro oblast národního parku, Tassili n'Ajjer, který sloužil jako předloha, jsou typické skalnaté formace, které jsou převážně vulkanického původu a z geologického hlediska tvořeny břidlicemi. Díky tomu se tyto náhorní plošiny vyznačují specifickými tvary.

Předlohou, byly jak snímky zachycené v samotné oblasti, tak také družicové snímky, které lépe odkrývají různorodý tvar skalnatých formací. Výsledné srovnání předlohy (vlevo) a generátoru vytvořeného v nástroji Houdini (vpravo) viz obrázek 7.10.



Obr. 7.10 Srovnání obrazové reference (vlevo) s výslednou horou vytvořenou v Houdini (vpravo).

7.4.1 Rozbor generátoru náhorních plošin - Houdini

Na rozdíl od písečných dun, které nejsou vyloženě běžným typem terénu, je vytváření skalnatých formací v nástroji Houdini daleko přímočařejší.

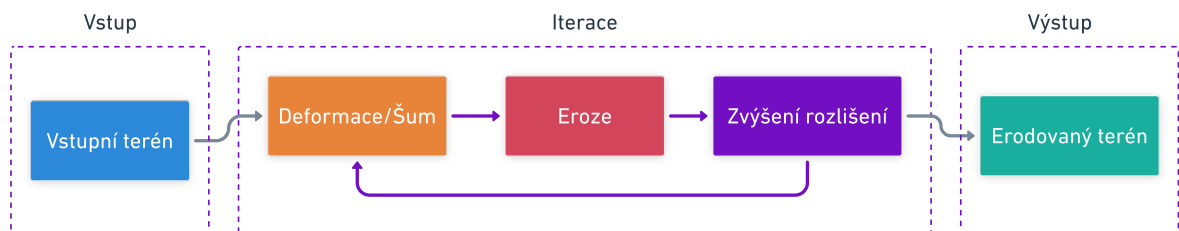
Základním vstupem vytvořeného generátoru je opět požadovaný tvar hory, který lze nakreslit přímo v Houdini pomocí *HeightField Draw Mask*. Díky tomu bylo možné vytvořit horu,

kteřá je „ušitá“ přímo na míru pro potřebu konkrétního záběru.

Tento tvar je následně pomocí *HeightField Blur* rozmazán, aby měla maska gradientní charakter. Jelikož byla maska vlivem rozmazání poměrně jednotvárná byl použit operátor *HeightField Distort*, který uniformní charakter masky rozčlenil. Zde je důležité zmínit, že tento postup není komutativní, tedy záleží na pořadí operátorů. Pokud bychom nejprve použili *HeightField Distort* následným rozmazáním bychom o detaily přišli.

Na obrazové předloze jsou patrné i menší skalnaté ostrůvky obklopující skálu. Ty však již nebyly tvořeny ručně nýbrž procedurálně pomocí *HeightField Mask Noise*. Dále byly veškeré vymaskované části vyvýšeny, přidán šum a aplikována první vrstva eroze (*HeightField Erode*), která určila hlavní vertikální členění.

Pro realistický výsledek je důležité pracovat iterativně a nesnažit se erozi terénu vytvořit pouze jedním operátorem eroze na samotném konci řetězce. Klíč k dosažení realisticky vypadajících pohoří spočívá ve vrstvení jednotlivých vrstev eroze na sebe. Obecný postup spočívá ve vytvoření terénu v relativně nízkém rozlišení, následném přidání šumu a celkovém zvrásnění terénu (např. pomocí *Heightfield Distort*) a až poté aplikací eroze. Je to z toho důvodu, že pokud by byl terén příliš hladký, voda by během eroze stékala po povrchu a neměla tendenci se do terénu zařezávat a tvořit koryta. Před erozí je tedy nutné zajistit, aby byl terén dostatečně členitý. Samotná eroze totiž terén také zhlazuje. Následně lze u terénu opět zvýšit rozlišení a tento celý postup opakovat znovu. Postup ilustruje obrázek 7.11.

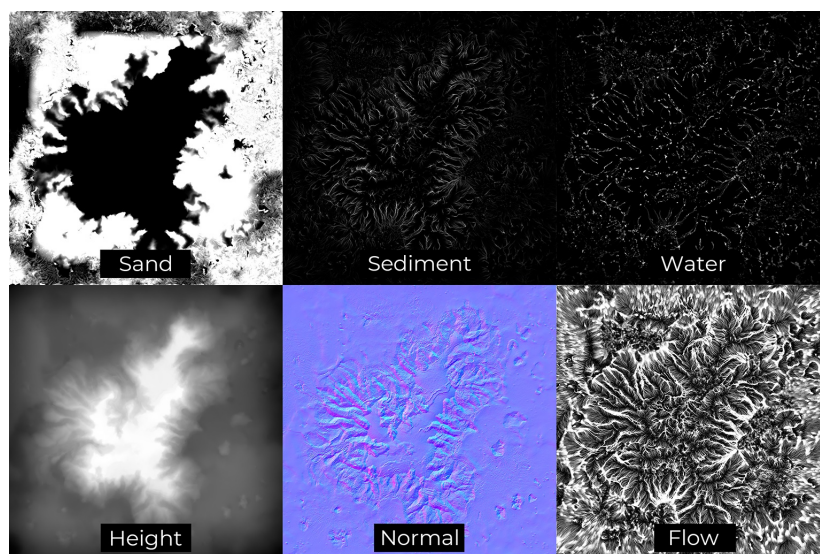


Obr. 7.11 Zázornění iterativního přístupu při erodování terénu.

Tímto postupem byl zformován základní tvar hory. Pomocí *HeightField Clip* byl poté vrchol skály zploštěn tak, aby odpovídal předloze. V obrazové referenci bylo v jistých místech rovněž patrné, členění do vrstev. Tohoto efektu bylo dosaženo pomocí operátoru *HeightField Terrace* a následném použití *Heightfield Distort*, který uniformní vzhled terasování narušil. Finálním krokem pak byla ještě jedna aplikace eroze, která vyhladila případné ostré přechody a celý terén jakoby propojila.

Pomocí tohoto generátoru bylo ve finále vytvořeno hned několik skalnatých formací, které

byly vkládány primárně do zadního plánu záběrů. Textury pro jednotlivé skály byly vytvořeny s využitím vygenerovaných masek a textur (obrázek 7.12) v nástroji Substance Painter.



Obr. 7.12 Ukázka jednotlivých textur vygenerovaných generátorem za účelem vytvoření finálního materiálu povrchu v nástroji Substance Painter.

7.5 Částicové efekty

Kromě samotného prostředí bylo potřeba vytvořit i částicové efekty, kterými bylo možné simulovat písek přenášený větrem. Zakomponování těchto dynamických prvků do scény napomohlo dynamice, a pusté prostředí pouště vizuálně oživilo. Kromě simulací písku bylo využito částicových efektů také v prostředí města, kde kouřové simulace dotvořily atmosféru industriálního města.

Pro vlastní kouřové a prachové efekty bylo opět využito nástroje Houdini v kombinaci se solverem třetí strany – Axiom. Přestože Houdini disponuje vlastním Pyro solverem, Axiom Solver je tzv. sparse GPU akcelerovaný solver, což znamená, že je pro tento účel efektivnější, a tudíž i rychlejší.

Po nastavení jednotlivých parametrů simulace bylo nutné zajistit, aby byla animace opakovací (*loopable*). Čili jinými slovy, počáteční a koncový stav simulace musel být totožný, aby při přehrávání ve smyčce nebyla patrná náhlá změna stavu. SideFX Labs, což je dodatečná sada nástrojů pro SideFX Houdini, v tomto směru nabízí operátor *Labs Make Loop*, který po zadání počátečního a koncového snímku zajistí simulaci plynulé opakování.

V rámci výsledného projektu byly otestovány dvě techniky renderování kouře v Unreal En-

ginu, které si vyžádaly odlišné přístupy při zpracování simulace. Tyto přístupy jsou popsány v následujících podkapitolách.

7.5.1 Problematika renderování kouře v Unreal engineu

Na úvod by ale bylo vhodné nastínit základní problematiku při renderování kouřových simulací v Unreal Engineu. Přesto, že má Unreal Engine svá specifika obecně lze následující postupy aplikovat i na ostatní herní enginey na trhu.

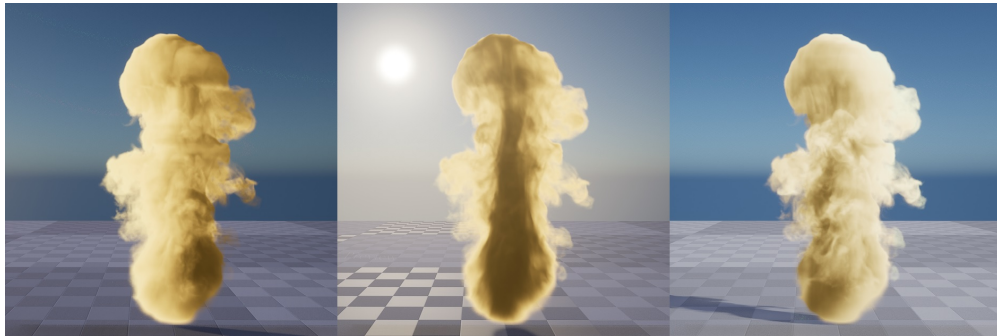
V real-time grafice existuje řada způsobů, jak vykreslovat animovaný kouř. Pro účely projektu však bylo využito tzv. flipbook animace, která spočívá v uložení simulace do jediné textury, která je rozdělená do políček, kde každé políčko představuje jeden snímek animace. V shaderu se poté, zjednodušeně řečeno, textura opět rozstříhá a jednotlivé snímky se jeden po druhém přehrají. Následně můžeme tento animovaný materiál namapovat na objekt a umístit do scény.

Problém, který zde nastane je však ten, že kouř se sice přehrává plynule bez zadrhnutí, nicméně již žádným způsobem nereflektuje nasvícení scény. S jakým nasvícením byl kouř vyrenderován a uložen do flipbooku s takovým je i nyní ve scéně. Jelikož tedy nechceme mít pro každý záběr jiný flipbook, který odpovídá nasvícení scény, je potřeba zvolit sofistikovanější metodu.

7.5.2 Zpracování simulace pro metodu Six Point Lighting

První metodou je využití tzv. šestibodového svícení. Myšlenka je taková, že kouř nasvítíme z každé strany samostatným světlem a následně animaci vyrenderujeme zvlášť pro každé světlo. Dostaneme tedy šest verzí kouřové animace, kde kouř osvětluje světlo pokaždé z jiné strany. Tyto jednotlivé verze lze poté v shaderu prolínat podle pozice světelného zdroje ve scéně. Ukázka výsledného kouřového efektu, který reflektuje pozici slunce ve scéně je znázorněna na obrázku 7.13.

Výhodou této metody je, že výsledný efekt vypadá poměrně věrohodně, nevýhodou však je náročná příprava simulace. V případě, že je animace dlouhá 64 snímků, znamená to, že pro každý snímek je potřeba 6 verzí z různým nasvícením. Pokud k tomu přičteme ještě alfa kanál, který určuje průhlednost kouře dostaneme $64 \times 6 + 64 = 448$ snímků. Částečnou pomocí zde je možnost uložení snímků do jednotlivých RGBA kanálů. Díky tomu se podaří 64 snímků dlouhou animaci vměstnat do jedné RGBA a jedné RGB textury. I tak se však



Obr. 7.13 Ukázka kouřového efektu v Unreal Engine, který reaguje na pozici světla ve scéně - světlo zleva, zezadu, zprava.

jedná o poměrně zdlouhavý a komplikovaný proces.

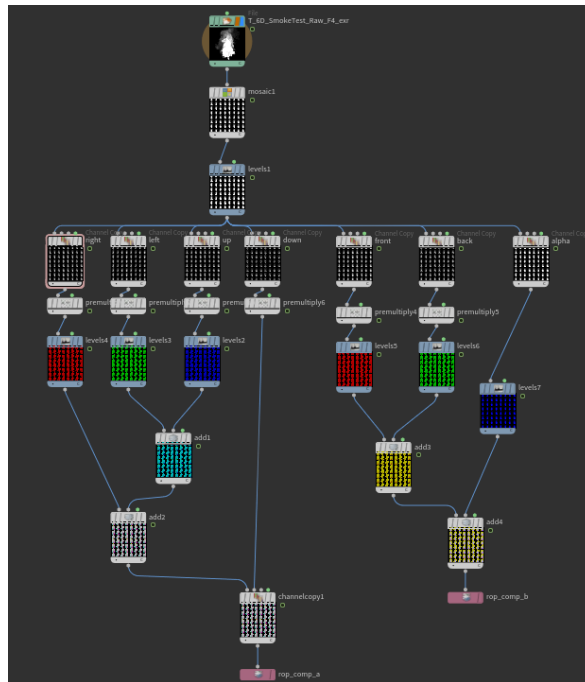
Houdini naštěstí dokáže značnou část kroků automatizovat. Například v případě samotného renderování kouře, lze jednotlivé verze nasvícení ukládat do jednotlivých *AOVs*, a uložit tak pro daný snímek veškeré varianty nasvícení do jednoho *.exr* souboru. Následně lze v *image network (COP)* využít operátor *Mosaic*, pomocí kterého je možné rozmístit jednotlivé snímky do podoby flipbooku a dále pomocí *Channel Copy* vyextrahovat jednotlivé varianty nasvícení a uložit do RGB respektive RGBA textur.

Ukázka *image network* sítě (viz obrázek 7.14), která se stará o načtení *.exr* animované sekvence, kterou dále konvertuje do flipbooku, rozdělí jednotlivé *AOVs* do RGBA kanálů a uloží výslednou dvojici flipbooků do dvou souborů. Tyto dva soubory obsahující animovanou simulaci již poté lze naimportovat do enginu, kde se dále zpracují v shaderu. Ukázka shaderu v Unreal Enginu viz obrázek 7.15.

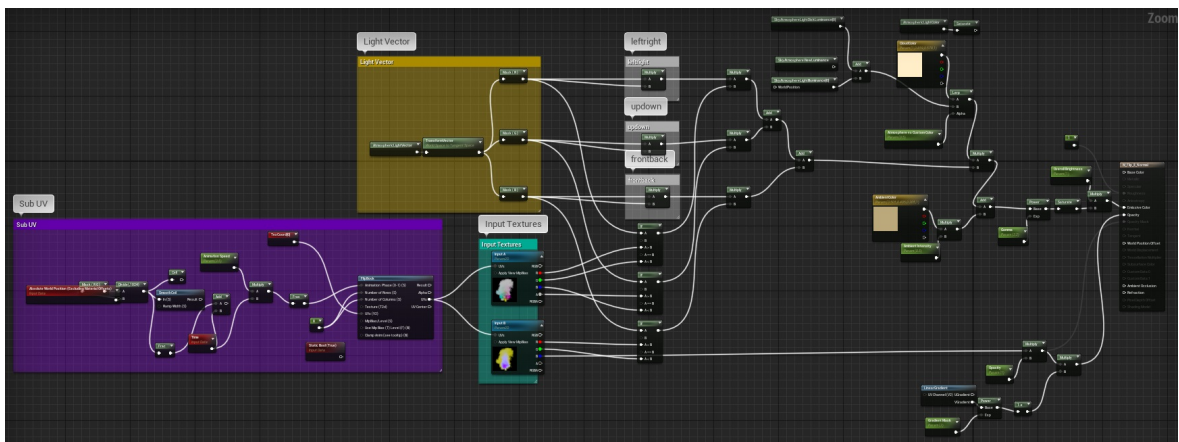
7.5.3 Zpracování simulace pro metodu 2D Raymarching

Druhá metoda je z hlediska přípravy daleko jednodušší. Vyžaduje totiž pouze jednu verzi kouře. Do enginu tedy stačí naimportovat jediný flipbook, který v jednotlivých snímcích představuje hustotu kouře. Veškerá logika a výpočet nasvícení se odehrává v samotném shaderu.

Výhodou tohoto přístupu je jeho jednoduchá příprava a univerzálnost použití. Nevýhodou je lehce vyšší výpočetní náročnost a jelikož se jedná pouze o aproximaci, tak také přesnost nasvícení výsledného kouřového efektu. Na druhou stranu, pro většinu situací, kde efekt není středobodem scény, je kvalita dostatečná.



Obr. 7.14 Ukázka COP sítě, která připravuje animovanou simulaci typu *Six Point Lighting* pro použití v Unreal Engineu.



Obr. 7.15 Ukázka UE shaderu, který zpracovává kouřovou simulaci typu *Six Point Lighting*.

7.6 Generátor mraků - Houdini

Pro potřeby filmu, byly rovněž vytvořeny dva typy generátorů mraků. První generátor využívá Houdini *Pyro Solver*, který je fyzikálně přesný, na druhou stranu ale umožňuje pouze omezenou míru kontroly nad výslednou podobou mraku. Tento typ ilustruje obrázek 7.16.

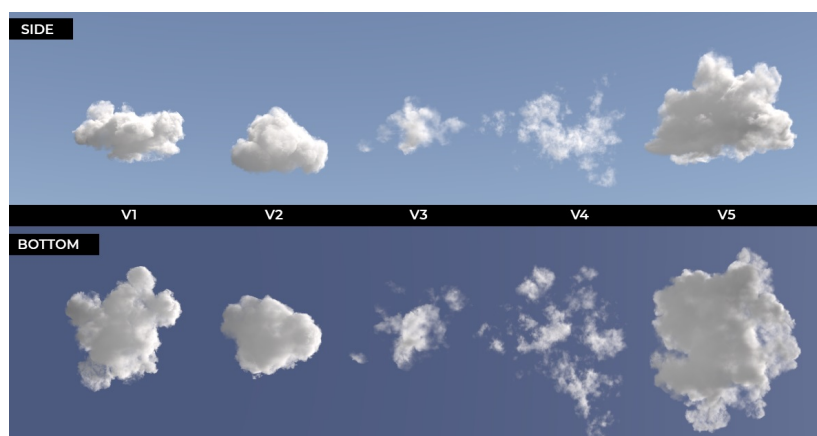
Druhý přístup využívá Houdini *SOP* v kombinaci s 3D šumovými funkcemi reprezentovanými operátorem *Cloud Noise*. Tento přístup není tak přesný, jako přístup s využitím *Pyro Solveru*, nicméně je možné mraky formovat do konkrétních tvarů. Je důležité zmínit, že ope-



Obr. 7.16 Ukázka mraku vytvořeného pomocí vlastního *Pyro* generátoru v Houdini.

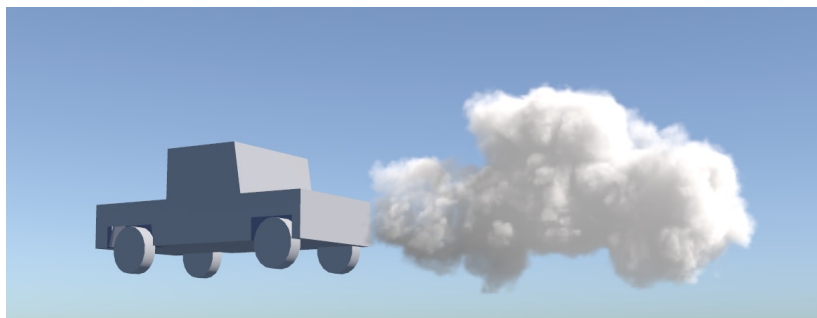
rátor *Cloud Noise* disponuje řadou parametrů a odvádí zde většinu práce. Klíčové ale bylo dosáhnout realistického tvaru, na který se poté šum aplikuje. Vytvořený generátor nabízí v tomto ohledu řadu parametrů, díky kterým lze kromě velikosti upravovat také členitost, hustotu v různých částech mraku nebo například ovlivnit míru zploštění spodní části, což je vlastnost typická především pro kupovitou oblačnost (*cumulus* mraky).

Pomocí generátoru tak lze vygenerovat mraky typu *Alto cumulus*, *Strato cumulus*, *Cumulus humilis*, *Cumulus mediocris* a *Cumulus congestus*.¹⁶ Ukázky různých hodnot parametrů generátoru ilustruje obrázek 7.17. Obrázek 7.18 poté ilustruje univerzálnost generátoru, který umožňuje formovat mraky do libovolných 3D tvarů.



Obr. 7.17 Ukázka různých mraků vytvořených pomocí vlastního *SOP* generátoru v Houdini.

¹⁶ROŠLAPIL, Ladislav. Základní druhy oblaků [online]. MetAmater.cz. 2021. Dostupné z: <http://www.metamater.cz/oblaka-srazky/zakladni-druhy-oblaku/>



Obr. 7.18 Ukázka vlastního tvaru mraku vytvořeného pomocí Houdini SOP.

8 Procedurální tvorba prostředí sci-fi města

Klíčovým a zároveň jedním z nejkompexnějších prostředí filmu je industriální sci-fi město zasazené do srdce pouště. Jelikož město ve filmu sehrává důležitou roli, bylo nutné ho vy-modelovat do detailů, které by umožnily kameru zasadit až dovnitř, mezi jednotlivé budovy. Finální podoba města ve filmu má rozlohu přes 12km^2 . Je tedy jasné, že vytvářet město ručně, bez jakýchkoliv procedurálních řešení by nebylo příliš reálné.

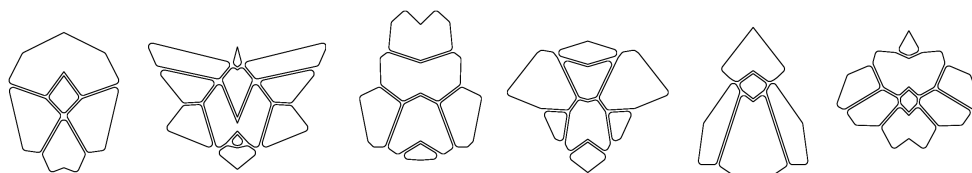
Město bylo vytvořeno převážně procedurálními technikami pomocí nástrojů Houdini a Substance Designer. Výsledné prostředí města ve filmu ilustruje obrázek 8.1.



Obr. 8.1 Prostedí industriálního města vytvořeného procedurálními technikami v Houdini a Substance Designeru. Vyrenderováno v reálném čase v Unreal Engine 5.

8.1 Generátor půdorysu sci-fi města - Substance Designer

Prvním krokem bylo definování tvaru a rozvržení města. I v tomto případě bylo využito procedurální řešení. Konkrétně byl pro tyto účely vytvořen generátor v Substance Designeru, který umožňoval generovat libovolný počet různých návrhů rozložení. Několik náhodně vybraných ukázek výstupu generátoru je možné vidět na obrázku 8.2.



Obr. 8.2 Ukázka několika vygenerovaných půdorysů města vytvořeného pomocí vlastního generátoru v Substance Designeru.

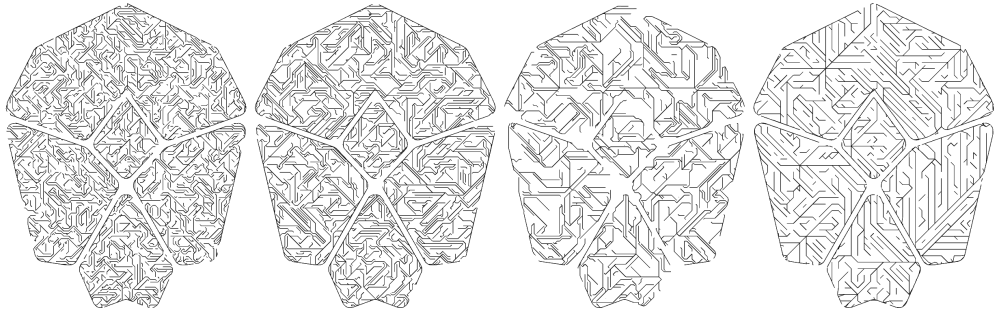
8.2 Generátor sci-fi města - Houdini

Textura s vygenerovaným půdorysem byla následně v podobě černobílé masky naimportována do Houdini, kde z ní byl pomocí operátoru *Trace* vyextrahován samotný tvar a převeden do geometrie.

Aby byla podpořena i vertikální členitost města, byly jednotlivé platformy (části) vyvýšeny v závislosti na obsahu jejich plochy. Větší plochy byly vyvýšeny méně, naopak plochy s menším obsahem byly vyzdviženy výše. Jednotlivé platformy byly poté mezi sebou propojeny procedurálně generovanými tunely.

Jelikož se jedná o futuristické industriální město, jeho infrastruktura byla navržena tak, aby svým rozvržením připomínala desku plošných spojů. K tomuto účelu byl vytvořen generátor, který dokáže na libovolném tvaru generovat tato spojení. Hustotu spojů včetně maximálních a minimálních délek lze upravovat pomocí vstupních parametrů generátoru. Ukázka vlivu různých hodnot parametrů generátoru viz obrázek 8.3.

Poté co byla vytvořená silniční a potrubní síť, byly do infrastruktury umístěny budovy. Budovy byly vytvořeny v odděleném procesu a jejich tvorbou se blíže zabývá kapitola 8.3. V první verzi byly budovy ve městě rozmístěny náhodně podél silniční a potrubní infrastruktury. Tento způsob se však příliš neosvědčil. Přestože byly jednotlivé budovy výškově i tvarově různorodé, rovnoměrným rozmístěním těchto budov se jakákoliv rozmanitost vytratila. Bylo tedy nutné přistoupit k pokročilejší metodě umístění budov, která by budovy rozmístila realističtějším způsobem.

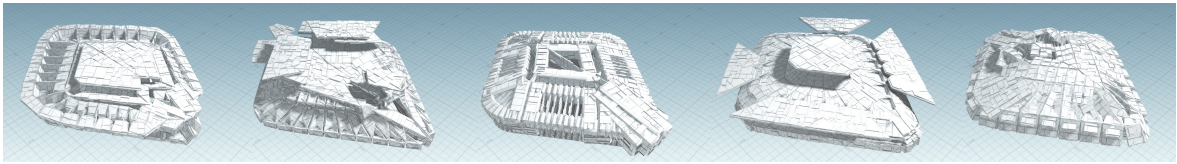


Obr. 8.3 Ukázka vlivu různých hodnot parametrů generátoru městské infrastruktury - Houdini.

Byl tedy vytvořen samostatný generátor, který bere v potaz výšku jednotlivých budov a dle vygenerované výškové mapy, umístí uje budovy do městské infrastruktury. Výsledkem je tak město, které je z profilu daleko členitější, neboť vysoké budovy jsou generovány pospolu a tvoří tak lokální centra, podobně jako to známe z dnešních velkoměst.

Některé, především nižší budovy, byly mezi sebou propojeny kabelovou sítí, která vizuálně přidala městu na komplexitě. S ohledem na modularitu celého řešení byl vytvořen samostatný generátor, který na základě několika vstupních parametrů jako např. minimální a maximální výška generování, délka a počet kabelů atd. propojoval jednotlivé budovy kabelovou sítí. Tento generátor funguje na principu vygenerování křivek nad městem, které se následně pomocí operátoru (*Ray*) připínají k nejbližším budovám pod sebou. Následně byl vytvořen efekt prověšení kabelu vlivem gravitace, který na základě délky kabelu (potažmo jeho hmotnosti), umožnil, že se kabel pod svou vahou prohne. Jak jde vidět, mnohdy se jedná o drobnosti, které v celkové komplexitě projektu naprosto zapadnou. Laik si dokonce může myslet, že se takové věci vytváří automaticky, nicméně mnohdy i takové detaily je nutné vytvořit ručně.

Centrem města je hlavní budova s velkou přistávací plochou na její střeše, schopnou pojmout loď o rozměrech i několika stovek metrů. Tato budova se od ostatních budov koncepčně liší, a proto byla vytvořena odlišným generátorem. Generátor vychází z tvaru platformy, na které se budova nachází a pomocí extrudování jednotlivých částí platformy generuje výslednou podobu budovy. Ukázky různých variant budovy viz obrázek 8.4. Na tomto příkladu lze vidět, že procedurální generátory lze snadno využít i k rapidnímu prototypování a vytváření návrhů. Generátor je totiž možné automatizovat a nechat ho vygenerovat stovky i tisíce takových obrázků, ze kterých si pak uživatel vybere ten, který se mu líbí nejvíce. Výsledný návrh může použít přímo, lehce ho poupravit, nebo může určit směr, kudy se dále vydat.



Obr. 8.4 Ukázky různých variant centrální budovy města - Houdini.

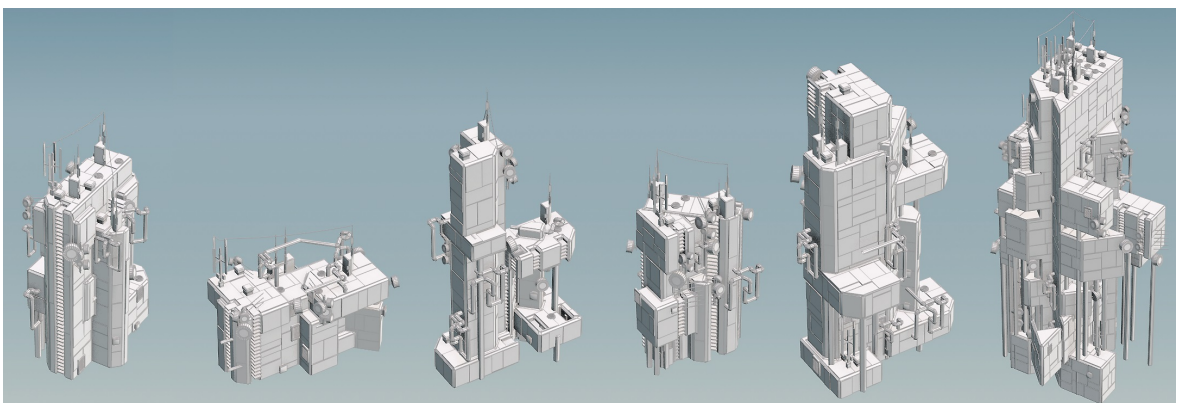
Na závěr byla kolem celého města vygenerovaná zeď. Vstupem tohoto generátoru je základní tvar města. Generátor následně pro tento tvar vygeneruje konvexní obal a stěny tohoto konvexního obalu slouží jako základ pro venkovní zdi. Aby zeď nepůsobila příliš jednotvárně byly podél ní vygenerovány podpěrné bloky. Celá zeď je krytá procedurálně generovanou střechou s výstupky v jednotlivých rozích a podepřená sloupy.

8.3 Generátor sci-fi budov - Houdini

Generátory budov jsou jednou z nejčastějších aplikací procedurálních řešení vůbec. Na toto téma tedy existuje řada publikací, které se touto problematikou zabývají a staly se tak předlohou pro tuto práci. Pro potřebu filmu bylo cílem vytvořit rozlehlé industriální město, čemuž musel odpovídat i design samotných budov [6] [7] [8].

Na začátku bylo nutné určit, jaký vizuální styl mají jednotlivé budovy mít. Od toho se poté odvíjelo nadefinování tvaru půdorysu budovy, velikost, určení – k jakým účelům je budova využívána a s tím přímo souviselo jaké prvky musí budova obsahovat.

Několik budov různých velikostí a tvarů vygenerovaných pomocí vytvořeného generátoru ilustruje obrázek 8.5.



Obr. 8.5 Ukázky nastavení různých parametrů při generování budov v Houdini.

Jelikož se jedná o poměrně komplexní strukturu s množstvím částí, které do sebe musí zapadat, bylo důležité postupovat iterativně – začít základními tvary a na nich postupně stavět

dál. Pro zajištění správné funkcionality bylo při vývoji důležité testovat různé varianty generování a zkoušet různou kombinaci parametrů, aby bylo ověřeno, že se generátor chová dle očekávání. Tento krok je obecně velmi důležitý, protože pokud bychom funkčnost netestovali během samotného vytváření, ale až na samotném závěru, mohlo by se stát, že by generátor v některých případech nefungoval zcela korektně. Případná chyba by se navíc mohla kumulovat, a v hraničních případech bychom dokonce museli některé části generátoru vytvářet znovu.

8.3.1 Rozbor generátoru sci-fi budov - Houdini

Nejprve byl vytvořen systém, který generuje základní tvar budovy, bez jakýchkoliv detailů. Tento tvar je složen z několika kvádrů ze zkosenými rohy, což dodává budovám jednoduchý industriální charakter.

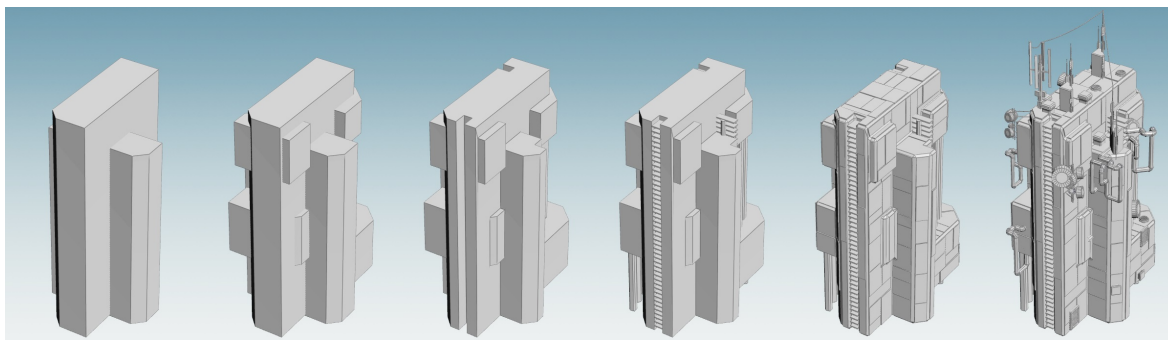
Dále byly na povrchu budovy vygenerovány body a na tyto body následně umístěny další bloky, které vzhledem připomínají malé moduly. Po tomto kroku byly v různých částech budovy vytvořeny zářezy, které byly dále vyplněny žebrováním. Pro tyto účely byl vytvořen vlastní generátor. Tímto byl v zásadě vytvořen základní tvar budovy a bylo tak možné přistoupit k přidávání detailů.

Nejprve byly pomocí generátoru panelů, který je založen na *Labs Lot Subdivision* z balíčku SideFX Labs, vytvořeny detaily na venkovních zdech. Tento operátor, dokáže rozdělit plochu na požadovaně velké panely, které byly následně vyextrudovány o náhodnou hodnotu ve specifikovaném rozsahu. Díky tomuto detailu byla rovná a nepřirozená linie zdí narušena a budova tak vypadala opticky mnohem zajímavěji.

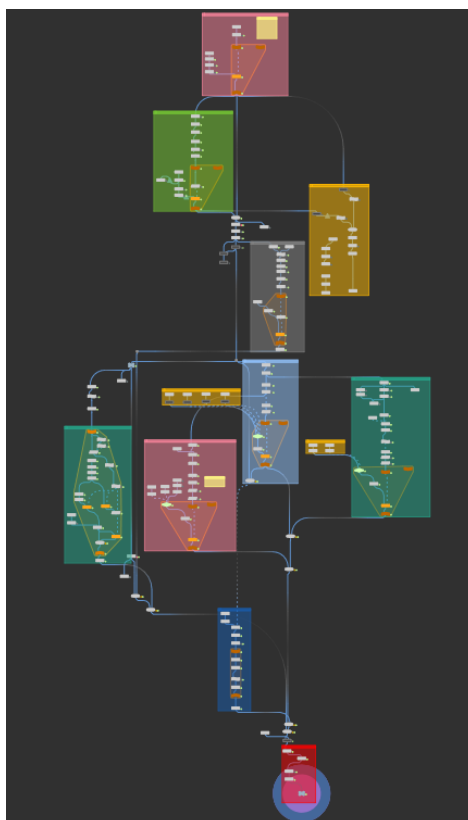
Dále byla vytvořena řada detailů (antény, větrací otvory, potrubní, kabely), které byly následně umístěny na budovu. V tomto případě bylo nutné vymaskovat potřebné části budovy a pomocí těchto masek procedurálně umístit jednotlivé objekty. Prvním krokem v tomto ohledu bylo vytvořit voxelovou reprezentaci objektu (*Labs Voxel Mesh*). Tímto vznikl prakticky vnější obal budovy. Dále bylo pomocí operátoru *Labs Measure Curvature* změřeno zakřivení objektu, které bylo použito jako maska k umístění dalších detailů.

Ukázka jednotlivých kroků při tvorbě generátoru na konkrétní budově viz obrázek 8.6.

Struktura výsledného grafu je vidět na obrázku 8.7. Z obrázku jsou patrné moduly, které jsou pro přehlednost v různobarevných rámečcích.



Obr. 8.6 Ukázka jednotlivých kroků při tvorbě generátoru budov v Houdini.



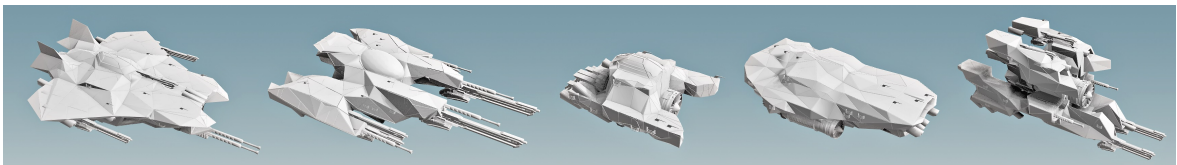
Obr. 8.7 Graf znázorňující výslednou strukturu generátoru budov - Houdini.

9 Procedurální tvorba vesmírných lodí

Ukázku různorodosti využitelnosti procedurálních technik pro generování objektů uzavírá generátor vesmírných lodí. Jak můžeme v řadě vědecko-fantastických filmů pozorovat, vesmírné lodě často nabývají různých tvarů a velikostí. Jelikož se mnohdy jedná o velmi komplexní objekty, které nejsou z pohledu vymodelování triviální, bylo původním záměrem vyhledat volně dostupné modely online. Ukázalo se však, že najít na internetu modely lodí, které by odpovídaly požadavkům, a navíc by byly ještě designově jednodušší, není snadné. Z těchto

důvodů bylo tedy přistoupeno k vytvoření vlastního procedurálního generátoru. Díky tomu jsou prakticky téměř všechny objekty ve filmu vytvořeny procedurálními technikami.

Generátor byl navržen tak, aby byl univerzální a bylo ho možné využít ke generování různých typů vesmírných lodí. Ve finální verzi generátoru lze ovlivňovat přes 80 parametrů, pomocí nichž lze vytvářet lodě až už průzkumné, vojenské nebo dopravní. Ukázky několika výstupů generátoru ilustruje obrázek 9.1. Ukázka jednoho modelu včetně textur a nasvícení v Unreal Engine viz obrázek 9.2.



Obr. 9.1 Ukázky různých lodí, které byly vygenerovány pomocí jediného generátoru v Houdini.



Obr. 9.2 Ukázka výsledné podoby jedné z lodí v Unreal Engine.

9.1 Generátor vesmírných lodí - Houdini

Prvním krokem při tvorbě generátoru bylo vytvoření základního tvaru. Ten byl vytvořen vygenerováním bodu v prostoru, do kterého byl následně umístěn náhodně vybraný objekt z balíčku předdefinovaných tvarů. Mezi tyto tvary patřily objekty jako hranoly, zkosené kvádry, válce apod. Po vložení prvního tvaru bylo na jeho povrchu vygenerováno několik dalších bodů a do těchto bodů umístěny další objekty. A takto se proces opakoval po stanovený počet iterací, který byl rovněž jedním z parametrů výsledného generátoru.

V dalším kroku bylo potřeba jednotlivé objekty spojit do jednoho uceleného objektu. K tomu bylo využito operátoru *Labs Voxel Mesh*. Tento operátor má ale ve výchozím nastavení

tendenci zaoblovat ostré rohy, především při nižším rozlišení sítě voxelů. Výsledkem by byl tvar, který by sice byl celistvý, ale také příliš zaoblený. Snahou však bylo vytvořit loď s ostře definovanými rysy, jaké můžeme znát ze současných vojenských stíhacích letounů s technologií *stealth*.¹⁷ Řešením je zvolení možnosti *Project Back to Original Geometry*. Díky této funkci se původně zaoblený tvar promítne zpět na původní objekt a tvar tak zůstane zachován.

V tomto kroku byla topologie objektu příliš detailní, a tak bylo potřeba počet polygonů redukovat. Bylo využito operátoru *Labs Poly Reduce* a počet polygonů snížen na nejnižší úroveň kdy ještě nedocházelo k přílišné deformaci objektu vlivem redukce.

Dále byl k trupu lodi přimontován motor. K tomuto účelu byl vytvořen sofistikovaný systém, který pomocí zadané pozice motoru upraví tvar trupu tak, aby motor nikde nevyčníval a byl přirozeně zasazen do draku lodi. Některé lodě měly motory dva, takže bylo nutné zajistit, aby systém fungoval i pro tyto případy. Následně byl vytvořen generátor otvorů pro nasávání motoru, který dle velikosti a sklonu polygonů v přední části trupu generoval tyto otvory pro nasávání.

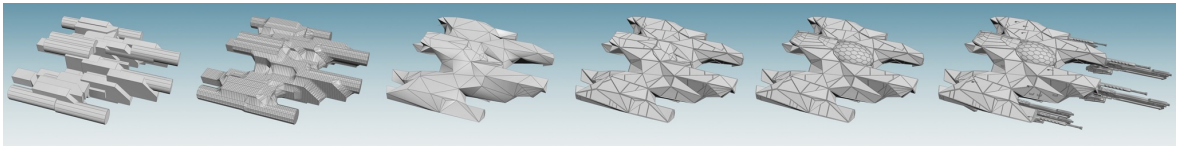
Dalším krokem bylo přidání kabiny. Byly vytvořeny dva systémy. První systém, generuje kabinu s rámováním vycházejícího z tvaru ploch panelů na trupu a druhý systém, který vytváří moderněji vypadající kabinu, připomínající kupoli složenou z hexagonální sítě. Oba systémy byly navrženy tak, aby do co největší míry pracovaly samostatně. Generátory samy naleznou optimální místo k umístění kabiny a následně v daném místě kabinu vytvoří.

Poté co byla dokončena finální podoba draku lodi bylo možné začít s přidáváním detailů. Prvním krokem bylo vygenerování jednotlivých panelů, které dohromady tvoří povrch trupu. Následovaly detaily jako například antény, indikátory rychlosti, závěsníky a také zbraňové systémy.

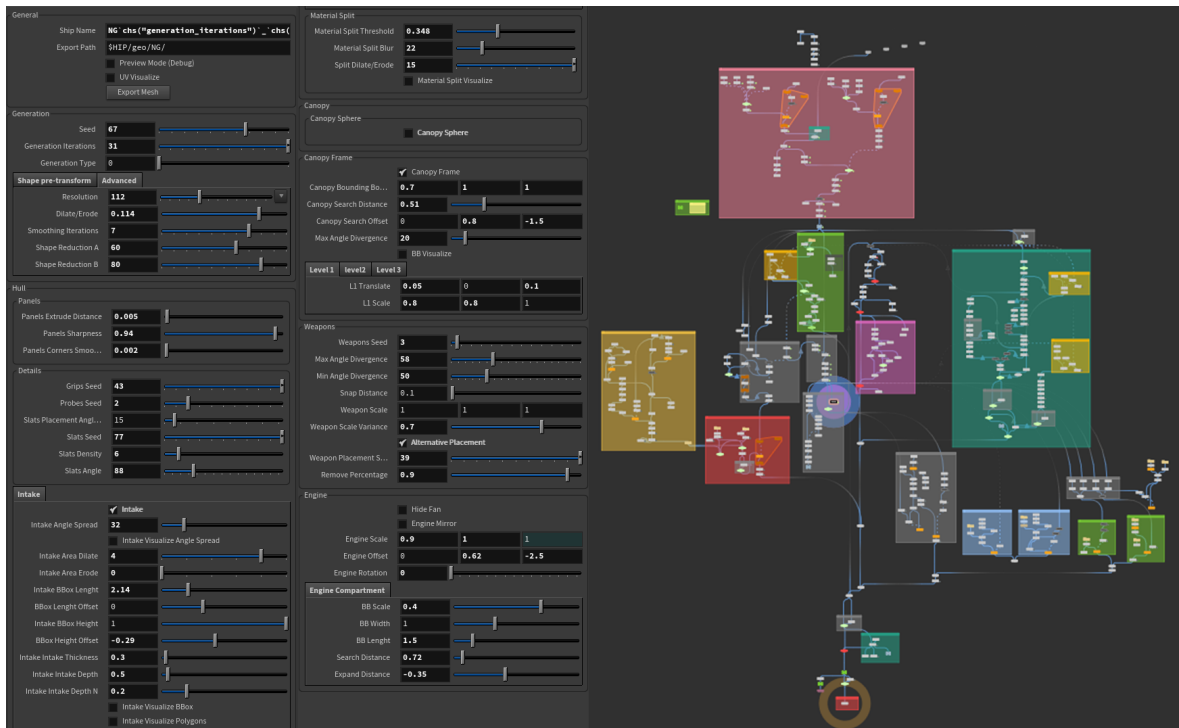
Ukázka jednotlivých kroků při tvorbě generátoru na konkrétním modelu lodi viz obrázek 9.3.

Přehled parametrů a struktura výsledného grafu generátoru lodí viz obrázek 9.4. Jednotlivé moduly jsou znázorněny v barevných rámečcích.

¹⁷Letouny s technologií „stealth“ (z angličtiny lze výraz přeložit jako „tajný“, „neviditelný“), jsou konstruovány takovým způsobem, aby šance na jejich odhalení pomocí radaru byla co možná nejmenší. Na rozdíl od konvenčních letadel, pro které je typický hladký aerodynamický tvar, je tvar těchto „neviditelných“ letadel složen ze zdánlivě změní různě nakloněných rovných ploch, tvořených většinou trojúhelníky nebo mnohoúhelníky. Takto navržená konstrukce má poté za cíl odrážet radarové vlny různými směry, tak aby se nevracely zpátky k radaru. Typickým a asi nejznámějším zástupcem takového typu letadla je Lockheed F-117 Nighthawk [11].



Obr. 9.3 Ukázka jednotlivých kroků při tvorbě generátoru lodí v Houdini.



Obr. 9.4 Přehled parametrů a graf znázorňující výslednou strukturu generátoru vesmírných lodí - Houdini.

10 Procedurální tvorba materiálů

Pro prostředí pouště bylo kromě skenovaných materiálů vytvořeno také několik procedurálně generovaných materiálů písku. Nejvíce materiálů však bylo vytvořeno pro potřeby sci-fi města a vesmírných lodí. Jednalo se především o různé materiály sci-fi panelů. Ke generování těchto materiálů bylo využito nástroje JSplacement v kombinaci se Substance Designerem. Ukázka několika takto vytvořených materiálů namapovaných na kouli viz obrázek 10.1.

Proces tvorby těchto materiálů spočíval ve vygenerování několika vzorů (výškových map) v JSplacement, které byly dále naimportovány do Substance Designeru. Tyto výškové mapy byly následně pomocí operátoru *Height Blend* zkombinovány do jedné a doplněny o další detaily vygenerované v Substance Designeru.



Obr. 10.1 Ukázka několika materiálů sci-fi panelů vytvořených pro části budov - JSplacement, Substance Designer (pro názornost jsou materiály namapovány na kouli).

V dalším kroku následoval proces vytváření povrchových nedokonalostí. Výstupy z nástroje JSplacement jsou totiž naprosto čisté, bez jakýchkoliv známek opotřebení. Přestože může být intuicí ponechat materiály co nejpěknější, vždy je nutné přemýšlet nad tím, kde a jak budou ve scéně použity. Jelikož byly tyto materiály určeny pro budovy industriálního města, idea dokonalých materiálů, bez jakýchkoliv povrchových nedokonalostí se s tímto příliš neshodovala. Materiály tak byly pomocí procedurálních generátorů záměrně ušpiněny. Přidány byly rovněž různé rýhy, škrábance, vrstvy rzi apod.

Přidávání těchto nedokonalostí je velice důležitým krokem při tvorbě jakýchkoliv materiálů, poněvadž díky nim vypadají materiály uvěřitelněji. V reálném světě se téměř nikdy neseznamujeme s povrchy, které by byly naprosto dokonalé. Vždy jsou nějakým způsobem ovlivněny okolím, ve kterém se nacházejí. Opačným extrémem je zase případ, kdy se umělec ve snaze o dosažení realisticky vypadajícího materiálu „nechá unést“ a materiál kompletně pokryje rží, nebo ho orazítkuje ze všech stran škrábanci. Vždy je důležité přemýšlet v kontextu – je konkrétní místo natolik exponované, a dostatečně odkryté aby se na něm nacházely škrábance? Nebo je pravděpodobné, aby se v tom či onom místě tvořila rez? Odpovědi na tyto otázky lze přitom najít většinou poměrně snadno – pozornou analýzou obrazové reference.

Na závěr byl materiál doplněn ještě o emisivní kanál, pomocí kterého bylo možné simulovat malé světelné zdroje, které rámuji jednotlivé části panelů. Jedná se o drobný detail, který ale činí materiál daleko zajímavějším. V shaderu lze totiž tento kanál různými způsoby animovat, či měnit jeho barvu a materiál tímto oživit.

ZÁVĚR

Cílem této práce bylo demonstrovat generování virtuálních prostředí za pomoci procedurálních technik. V prvních kapitolách se práce věnuje popisu procedurálních přístupů a jejich porovnáním s ostatními technikami. Základem každého procedurálního systému je vhodný návrh, který umožní rychlé a snadné zásahy do systému generování. Díky tomu, že je modulární, lze jeho části libovolně využívat a kombinovat i v dalších budoucích projektech. Snahou bylo také předvést, že procedurální neznamena nutně „náhodně generované“. Vhodným návrhem lze totiž zajistit dostatečnou „uměleckou kontrolu“ při zachování všech výhod, které s sebou procedurální techniky přinášejí.

Tyto teoretické poznatky byly aplikovány při tvorbě jednotlivých generátorů pro použití ve filmu „Poušť“, který demonstruje široké využití procedurálních technik v kombinaci s real-time technologiemi. Veškerá prostředí ve filmu byla generována procedurálními technikami v nástrojích Houdini a Substance Designer. Pro film bylo celkově vytvořeno devět různých variant generátoru pouští, generátor sci-fi města s veškerou infrastrukturou a procedurálně generovanými budovami. Dále byl vytvořen poměrně univerzální generátor vesmírných lodí s bezmála 80 parametry, pomocí kterého lze vytvářet lodě různých typů a velikostí. Vytvořeny byly také procedurální materiály pro většinu povrchů objektů ve scénách. Co se týče atmosferických efektů, v tomto ohledu byly vytvořeny dva generátory mraků, jenž byly postaveny na odlišných technologiích – jeden s využitím simulace a druhý s využitím šumových funkcí. V neposlední řadě byly vytvořeny také částicové efekty, pomocí nichž byly ve filmu simulovány efekty jako jsou prach nebo kouř.

Konkrétně byl film renderován v „early access“ verzi Unreal Engine 5, který vyšel v průběhu psaní této práce. Díky zpětné kompatibilitě bylo možné scény, které vznikly před vydáním nové verze, konvertovat a naimportovat do Unreal Engine 5. Tato verze je totiž naprosto revoluční z pohledu zpracování renderovaných 3D objektů. Pomocí systému pro virtualizaci geometrie „Nanite“, dokáže zpracovávat a vykreslovat řádově násobně detailnější geometrii, než bylo doposud možné. Daný systém tak prakticky odbourává limity na počty polygonů nebo tzv. „draw calls“, které byly v real-time grafice doposud značným omezením. Jedná se proto o naprosto přelomovou technologii, která nepochybně navždy změní podobu real-time grafiky.

Využitím těchto technologií bylo možné vytvořit prostředí města do takových detailů, jaké dosud real-time technologie neumožňovaly. Objekty ve městě čítají dohromady stovky mili-

onů polygonů, a přesto vykreslování probíhalo naprosto plynule i na průměrném notebooku. Kombinace procedurálních technik s real-time grafikou se ukázala jako naprosto ideální řešení. Nejen, že bylo možné jednotlivé generátory rychle jakkoliv a kdykoliv upravovat (což je především v oblasti VFX, velice důležitý bod), ale zásluhou Unreal Engine se změna promítla do scény okamžitě. Díky tomu je celý proces tvorby daleko rychlejší, efektivnější a zároveň také kreativnější.

Toto zrychlení má pochopitelně přínos nejen časový, ale také finanční. Renderovat komplexní scény se stovkami milionů polygonů už není výsadou render farm, nýbrž s Unreal Enginem tento úkol zvládne i lehce výkonnější desktop. Tento fakt otevírá dveře malým studiím či jednotlivcům, kteří budou schopni produkovat vysoce kvalitní digitální prostředí s minimálními náklady. Do budoucna lze očekávat rozmach těchto technologií, které umožní i malým filmařům přenášet své odvážné nápady na velká plátna.

SEZNAM POUŽITÉ LITERATURY

- [1] BONATTI, Luca, Marco ROLANDI, Julien BOLBACH a Kai WOLTER. DMP without DMP, full-CG environments for The Lion King. In: ACM SIGGRAPH 2019 Talks [online]. ACM, 2019. ISBN 9781450363174. Dostupné z: doi:10.1145/3306307.3328195
- [2] OKUN, Jeffrey A. a Zwerman, Susan. The VES Handbook of Visual Effects: Industry Standard VFX Practices and Procedures. Focal Press, 2010. ISBN 978-0-240-81242-7
- [3] QUINN, Kevin. Procedural Rocks [online]. Artstation, copyright 2021 [cit. 2021-01-02]. Dostupné z: <https://www.artstation.com/artwork/L2aD00>
- [4] VON ÜBEL, Max. Best Photogrammetry Software in 2021 [online]. All3DP, copyright 2021 [cit. 2021-02-05]. Dostupné z: <https://all3dp.com/1/best-photogrammetry-software/>
- [5] HARISOVA, Ellie. Houdini Engine: Directed Procedural Workflow in Houdini and Unity [online]. 80 Level, copyright 2021 [cit. 2021-01-07]. Dostupné z: <https://80.lv/articles/houdini-engine-directed-procedural-workflow-in-houdini-and-unity/>
- [6] MÜLLER, Pascal, Peter WONKA, Simon HAEGLER, Andreas ULMER a Luc VAN GOOL. Procedural modeling of buildings. ACM Transactions on Graphics [online]. 2006, 25(3), 614–623. Dostupné z: doi:10.1145/1141911.1141931
- [7] KELLY, Tom a Peter WONKA. Interactive architectural modeling with procedural extrusions. ACM Transactions on Graphics [online]. 2011, 30(2), 1–15. Dostupné z: doi:10.1145/1944846.1944854
- [8] CHEN, Guoning, Gregory ESCH, Peter WONKA, Pascal MÜLLER a Eugene ZHANG. Interactive procedural street modeling. ACM Transactions on Graphics [online]. 2008, 27(3), 1–10. Dostupné z: doi:10.1145/1360612.1360702
- [9] KADNER, Noah. The Virtual Production Field Guide [online]. Epic Games, copyright 2019 [cit. 2021-02-03]. Dostupné z: <https://www.unrealengine.com/en-US/vpfieldguide>
- [10] ROŠLAPIL, Ladislav. Základní druhy oblaků [online]. MetAmater.cz. 2021 [cit. 2021-07-06]. Dostupné z: <http://www.metamater.cz/oblaka-srazky/zakladni-druhy-oblaku/>

- [11] KUSALA, Jaroslav. „Neviditelné“ letouny [online]. Ministerstvo obrany ČR, copyright 2021 [cit. 2021-06-05]. Dostupné z: https://www.army.cz/images/id_8001_9000/8753/radar/f14.htm
- [12] MAGEE, Robert, Houdini Foundations for Film, TV & Gamedev. SideFX Software, 2018. ISBN 978-1-7753338-0-7. Dostupné také z: https://www.sidefx.com/media/uploads/tutorial/foundations_gdc2018/houdini_foundations.pdf

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

- AOV *Arbitrary Output Variable*, vrstva/pass, do které lze při renderování ukládat různá data. Např. hloubkovou mapu (ZDepth), normály, cryptomatte atd.
- CPU z angl. *Central Processing Unit*, centrální procesorová jednotka
- GPU z angl. *Graphics Processing Unit*, grafický procesor
- UE Unreal Engine
- VFX z angl. *Visual Effects*, vizuální efekty

SEZNAM OBRÁZKŮ

1.1	Využití procedurálního generování ve filmu The Lion King (2019, R: Jon Favreau).	14
1.2	Město vytvořené s pomocí procedurálních technik ve filmu Doctor Strange (2016, R: Scott Derrickson).	14
2.1	Ukázka ručně vymodelovaného 3D modelu skály z filmu The Lion King (2019).	18
2.2	Ukázka procedurálně generovaného 3D modelu skály.	19
2.3	Ukázka 3D modelu zrekonstruované skály pomocí fotogrammetrie.	21
2.4	Ukázka různých hodnot parametru procedurálně generovaného materiálu.	23
2.5	Ukázka různých verzí procedurálního materiálu (změna hodnoty „random seed“).	23
5.1	Generátor metropolitních měst vytvořen v nástroji Substance Designer.	28
5.2	Generátor středověkého města vytvořen v nástroji Substance Designer.	29
5.3	Generátor terénu pokrytého lesy vytvořen v nástroji Substance Designer.	29
6.1	Výsledné prostředí industriálního města z filmu vytvořeno za pomoci procedurálních technik. Renderováno v reálném čase v Unreal Engine 5.	31
7.1	Ukázka části obrazové předlohy použité k tvorbě generátorů.	32
7.2	Pouštní duny vytvořené procedurálními technikami v Houdini a Substance Designeru. Vyrenderováno v reálném čase v Unreal Engine 5.	33
7.3	Ukázka jednotlivých sub-generátorů písečných formací, vytvořených v nástroji Houdini.	34
7.4	Ukázka jemného zvrásnění dosaženo pomocí simulace eroze v nástroji Houdini.	34
7.5	Ukázka výsledného povrchu pouště o rozloze 25km ² vytvořeného kombinací jednotlivých sub-generátorů a jeho výsledný graf v nástroji Houdini.	35
7.6	Ukázka výsledných dun generátoru, které jsou generovány na základě jednoduchých vstupů (jednoduchá schémata v rozích) - Substance Designer.	36
7.7	Graf generující pouštní duny v Substance Designeru.	37
7.8	Ukázka vygenerovaných textur různých typů povrchových nerovností - Substance Designer.	37
7.9	Srovnání scény před a po aplikaci textur simulující drobné povrchové nerovnosti.	38

7.10	Srovnání obrazové reference (vlevo) s výslednou horou vytvořenou v Houdini (vpravo).	38
7.11	Zázornění iterativního přístupu při erodování terénu.	39
7.12	Ukázka jednotlivých textur vygenerovaných generátorem za účelem vytvoření finálního materiálu povrchu v nástroji Substance Painter.	40
7.13	Ukázka kouřového efektu v Unreal Engine, který reaguje na pozici světla ve scéně - světlo zleva, zezadu, zprava.	42
7.14	Ukázka COP sítě, která připravuje animovanou simulaci typu <i>Six Point Lighting</i> pro použití v Unreal Enginu.	43
7.15	Ukázka UE shaderu, který zpracovává kouřovou simulaci typu <i>Six Point Lighting</i>	43
7.16	Ukázka mraku vytvořeného pomocí vlastního <i>Pyro</i> generátoru v Houdini.	44
7.17	Ukázka různých mraků vytvořených pomocí vlastního <i>SOP</i> generátoru v Houdini.	44
7.18	Ukázka vlastního tvaru mraku vytvořeného pomocí Houdini <i>SOP</i>	45
8.1	Prostředí industriálního města vytvořeného procedurálními technikami v Houdini a Substance Designeru. Vyrenderováno v reálném čase v Unreal Engine 5.	45
8.2	Ukázka několika vygenerovaných půdorysů města vytvořeného pomocí vlastního generátoru v Substance Designeru.	46
8.3	Ukázka vlivu různých hodnot parametrů generátoru městské infrastruktury - Houdini.	47
8.4	Ukázky různých variant centrální budovy města - Houdini.	48
8.5	Ukázky nastavení různých parametrů při generování budov v Houdini.	48
8.6	Ukázka jednotlivých kroků při tvorbě generátoru budov v Houdini.	50
8.7	Graf znázorňující výslednou strukturu generátoru budov - Houdini.	50
9.1	Ukázky různých lodí, které byly vygenerovány pomocí jediného generátoru v Houdini.	51
9.2	Ukázka výsledné podoby jedné z lodí v Unreal Enginu.	51
9.3	Ukázka jednotlivých kroků při tvorbě generátoru lodí v Houdini.	53
9.4	Přehled parametrů a graf znázorňující výslednou strukturu generátoru vesmírných lodí - Houdini.	53

10.1	Ukázka několika materiálů sci-fi panelů vytvořených pro části budov - JSplacement, Substance Designer (pro názornost jsou materiály namapovány na kouli).	54
------	---	----