

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Fošebauer** Jméno: **Richard** Osobní číslo: **465974**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra počítačové grafiky a interakce**
Studijní program: **Otevřená informatika**
Specializace: **Počítačové hry a grafika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Fotorealistický konfigurátor vzhledu bazénu

Název bakalářské práce anglicky:

Photorealistic Swimming Pool Configurator

Pokyny pro vypracování:

Zmapujte metody a nástroje vhodné pro fotorealistické zobrazování vody se specifickým zaměřením na vzhled venkovních bazénů. Vyberte existující nástroj pro realistické zobrazování takových scén. Vytvořte konfigurovatelný model bazénu a jeho blízkého okolí. Zohledněte zejména možnost záměny povrchu bazénu, tedy různé typy materiálu stěn a dna bazénu (nerez, různé varianty fólie, plast). Umožněte rovněž konfiguraci lemu bazénu a okolního povrchu (dlažba, dřevo, trávnik, apod.). Konfigurace umožní specifikovat osvětlení scény (jasný letní den, zataženo, podvečer, apod.). Vytvořte aplikaci, která umožní konfigurovat finální vzhled scény a následně provede fotorealistickou syntézu vzhledu vytvořené scény z několika předdefinovaných pohledů. Implementaci otestujte vytvořením obrázků pro nejméně patnáct různých konfigurací. Vyhodnoťte časovou náročnost výpočtu a navrhněte případné optimalizace pro snížení výpočetního času.

Seznam doporučené literatury:

- [1] Pharr et al. Physically Based Rendering 3rd edition. Morgan Kaufmann 2016.
- [2] Aleš Koblížek. Realistické zobrazování vodních ploch. Diplomová práce, ČVUT FEL, 2020.
- [3] Renolit Alkorplan. Product specifications.
- [4] PBRT scenes (van der Rohe's Barcelona Pavilion, villa: Modern indoor environment). <https://www.pbrt.org/scenes-v3.html>

Jméno a pracoviště vedoucí(ho) bakalářské práce:

doc. Ing. Jiří Bittner, Ph.D. Katedra počítačové grafiky a interakce

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

Datum zadání bakalářské práce: **14.09.2022**

Termín odevzdání bakalářské práce: **26.05.2023**

Platnost zadání bakalářské práce: **19.02.2024**

doc. Ing. Jiří Bittner, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

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

III. PŘEVZETÍ ZADÁNÍ

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

Datum převzetí zadání

Podpis studenta

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra počítačové grafiky a interakce

Fotorealistický konfigurátor vzhledu bazénu

Richard Fošenbauer

Školitel: doc. Ing. Jiří Bittner, Ph.D.
Květen 2023

Poděkování

Děkuji mému vedoucímu práce doc. Ing. Jířímu Bittnerovi, Ph.D. za jeho rady, trpělivost a navádění správným směrem.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.

V Praze, 26. května 2023

Abstrakt

Tato práce probírá problematiku fotorealistického zobrazování vodní hladiny bazénu. Dále probere různé renderovací techniky, vhodné pro zobrazení scény s bazény, a výběr vhodného rendereru, který využívá jednu z těchto technik. Vyrenderované modely bazénů s různou texturou a osvětlením lze použít například jako referenci při nákupu bazénů, jelikož ne vždy dlaždice vypadají dobře za každého počasí a osvětlení. V práci je uveden postup modelování bazénu v Blenderu a srovnání vyrenderovaných snímků, včetně statistik. Dva různé rendery jsou porovnány z hlediska jejich schopnosti vykreslovat kaustiky. Dále jsou představeny a srovnány různé konfigurátory aut po nichž následuje popsání tvorby aplikace pro konfiguraci bazénu.

Klíčová slova: renderování, renderování vody, kaustiky ve vodě, renderování kaustik, bazén, Blender, LuxCoreRenderer, Cycles, konfigurátor bazénu, HTML aplikace

Školitel: doc. Ing. Jiří Bittner, Ph.D.
E-421,
Karlovo náměstí 13,
12000 Praha 2

Abstract

This work discusses the problematic of photorealistic rendering of water in pools. Then it goes through various rendering techniques, which are suitable for rendering scenes with pools with water, and choice of appropriate renderer using one of the techniques. Rendered models of pools with different textures and lightning can be used as a reference when purchasing a pool since the pool tiles does not always look good under different lightning and weather. In this work is also stated the process of modeling the pool in Blender and comparison of rendered images with statistics. Two different renderers are compared for their ability to render caustics: Cycles and LuxCore. Furthermore, various car configurators are presented and compared, followed by a description of the creation of an application for configuring a pool.

Keywords: rendering, water rendering, water caustics, caustic rendering, pool, Blender, LuxCoreRenderer, Cycles, pool configurator, HTML application

Title translation: Photorealistic pool configurator

Obsah

1 Úvod	1
2 Výběr technologií	3
2.1 Ostatní konfiguratory	3
2.2 Simulace a zobrazování vodní hladiny	5
3 Výběr rendereru	9
3.1 Srovnání	9
3.2 Blender a LuxCoreRenderer	9
4 Modelování bazénu	11
4.1 Základní tvar	11
4.2 Jednotlivé konfigurace	12
5 Rendering scény	13
5.1 Vykreslování různých konfigurací	13
5.2 Tabulky a výsledné rendery	16
5.3 Srovnání Cycles a LuxCore rendererů	19
6 Webová aplikace	21
6.1 Tvorba aplikace	21
6.2 Finální vzhled aplikace	22
7 Závěr	23
Literatura	25

Obrázky

2.1 Kofigurátor ŠKODA.	3
2.2 Kofigurátor BMW.	4
2.3 Kofigurátor Mercedes-Benz.	4
2.4 Aplikace displacement mapy v Blenderu	5
2.5 Ilustrace BSDF. Ve vrchní polovině obrázku je odraz světla popsán BRDF, ve spodní polovině naopak BTDF.	6
2.6 Obousměrný path tracing, paprsky vychází ze zdroje světla a od kamery zároveň	8
4.1 Scéna v Blenderu, bez textur ...	12
5.1 Jednoduchý bazén, Cycles	13
5.2 Cycles a LuxCore, složitější model.	14
5.3 Cycles den, 512 Samples - 23 min vlevo, 32 min vpravo	14
5.4 Cycles noc, 512 Samples - 28 min vlevo, 31 min vpravo.	14
5.5 LuxCore renderer - 42 min.	15
5.6 Kaustiky v Cycles rendereru. ...	15
5.7 Tyrkysová mozaika, slunečný den	16
5.8 Modrozelené kachličky, slunečný den	17
5.9 Modrobílý vzor, nižší viditelnost	18
5.10 Tyrkysová mozaika, noční osvětlení	19
5.11 Modrobílé kachličky, noční osvětlení	19
5.12 Denoiser v Cycles - 256 vzorků.	20
5.13 LuxCore, 512 samples, bidirectional path tracing, denoised.	20
6.1 Raná verze aplikace.	21
6.2 Finální vzhled aplikace.	22
6.3 Finální vzhled aplikace.	22

Tabulky

2.1 Tabulka indexů refrakce různých materiálů	7
5.1 Časy renderů pro animace finálního bazénu.	16
5.2 Časy renderů pro různé konfigurace při slunečném počasí.	16
5.3 Časy renderů pro různé konfigurace při slunečném počasí.	17
5.4 Časy renderů pro různé konfigurace za horší viditelnosti.	17
5.5 Časy renderů pro různé konfigurace za horší viditelnosti.	18
5.6 Časy renderů pro různé konfigurace v noci.	18
5.7 Časy renderů Cycles a LuxCore. LuxCore na 512 vzorků byl renderován pouze pomocí CPU. ...	20



Kapitola 1

Úvod

Stavba bazénu je poměrně velká investice, tudíž si kupující chce být jistý, že si vybral správně. Problémem je, že většina firem nabízející stavbu bazénů, je často mají vyfocené pouze za příznivých podmínek. Zákazník si většinou nemůže předem zkontrolovat vzhled dlaždiček za různého počasí.

Proto se ve své práci zaměřím na vytvoření programu, kde si bude uživatel moct vybrat vzhled bazénu a počasí. Program mu následně ukáže krátkou animaci, jak uživatelem zvolený bazén bude vypadat.

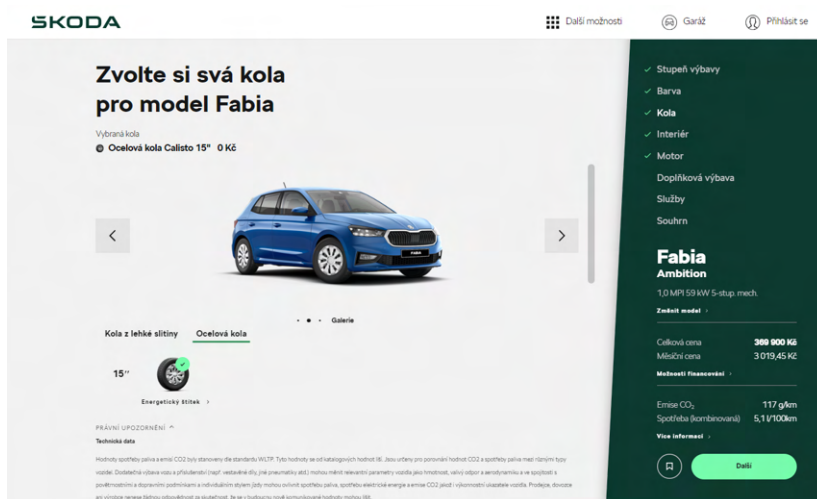
Nejprve, v kapitole 2, porovnám několik konfigurátorů aut, ze kterých jsem čerpal inspiraci při tvorbě aplikace. Dále proberu problematiku a metody zobrazování vody na počítači. Pro vytvoření animací jsem si potřeboval zvolit správný 3D modelářský program a renderer. Tímto výběrem se zabývám v Kapitole 3. Poté následuje Kapitola 4, kde popisuji svou práci v Blenderu, čili samotné modelování bazénu a volbu textur. Rendering scény s bazénem, včetně časových statistik, probírám v Kapitole 5. V téže kapitole srovnávám různé renderery, které jsem použil. V 6. kapitole představím a porovnám několik fotorealistických konfigurátorů aut, po nich následuje tvorba vlastní aplikace pro konfiguraci bazénu. Závěrem práce, v poslední kapitole, shrnu mou práci a zamyslím se, jak by se dala má aplikace vylepšit.

Kapitola 2

Výběr technologií

2.1 Ostatní konfigurátory

Abych mohl vytvořit aplikaci, potřeboval jsem najít inspiraci. Tu jsem našel u kofigurátorů aut. Jedná se o podobný koncept jako u konfigurátoru bazénu: v obou případech chce zákazník nejprve vidět, jak daný produkt bude vypadat, než si ho za nemalou částku zakoupí. Jak auto, tak bazén jsou poměrně velké investice, u kterých se předpokládá dlouhá doba užívání.

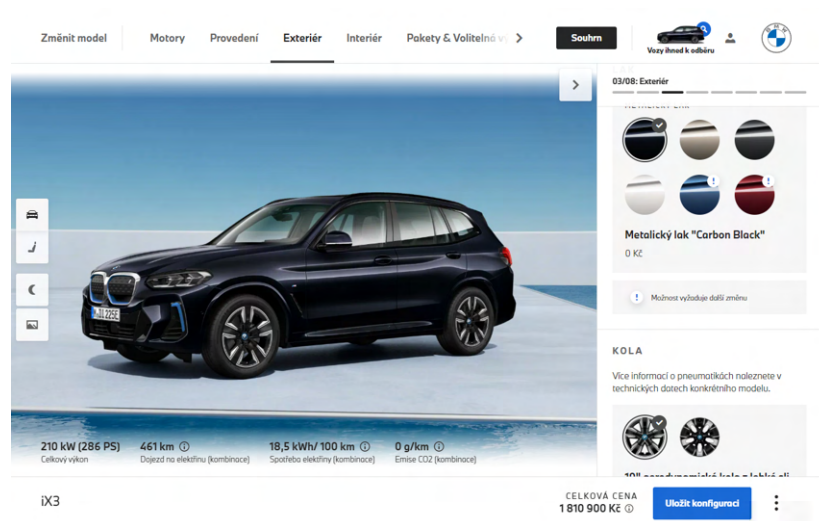


Obrázek 2.1: Kofigurátor ŠKODA.

Nejprve jsem zkoumal kofigurátor ŠKODA [8]. Při změně materiálu a doplňků, se mění předrenderovaný obrázek auta. Aplikace trochu upřednostňuje moderní design před uživatelskou přívětivostí 2.1. Vytkl bych jen, že poměrně dlouho trvá načítání, když uživatel přepíná mezi jednotlivými kategoriemi.

Druhý zkoumaný konfigurátor je od BMW. Narozdíl od ŠKODA konfigurátoru si uživatel může s autem otáčet, jelikož nepoužívají pouze statický obrázek auta. Pro exteriér se zdá, že aplikace vyrenderuje jeden obrázek na zhruba každých 10 stupňů. Interiér je předrenderovaná 3D scéna, kde uživatel může plynule otáčet kamerou. Design aplikace je příjemnější a přehlednější, načítání mezi kategoriemi není tolik intrusivní 2.2. Další z plusů

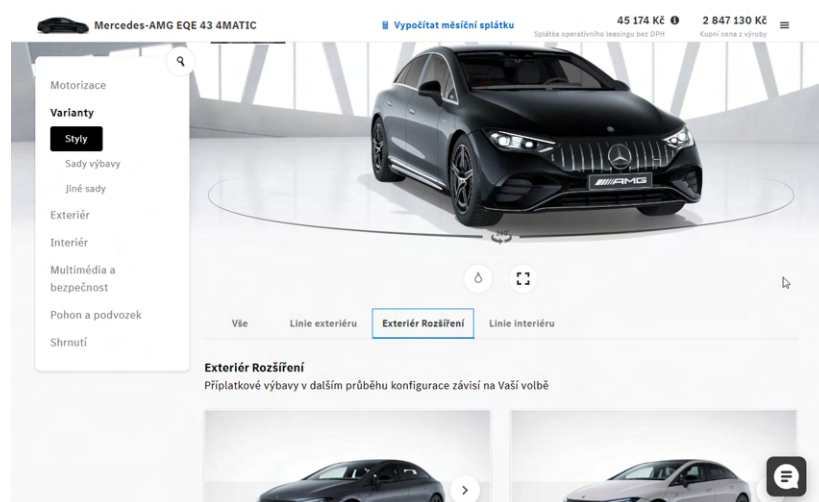
BMW konfigurátoru je možnost uložit si vybranou konfiguraci na později.



Obrázek 2.2: Kofigurátor BMW.

Jako další jsem našel konfigurátor aut Volkswagen [10]. Ten mě velice zklamal, neboť má pouze předrenderovaných 12 pohledů na exteriér auta, takže s ním zákazník může otáčet. Nelze měnit barva ani doplňky. Zkusil jsem konfigurátor od Mercedes-Benz [11]: zde s autem lze také otáčet, stejně plynule jako u BMW. Lze vybrat spoustu variací: od kol, přes světlomety, po interiér. Konfigurace lze také uložit na později. Co se týče uživatelské přívětivosti, tak je trochu nekonzistentní, že některé věci vybírám ve sloupci na straně obrazovky a jiné pod vyrenderovaným obrázkem auta 2.3.

Z těchto konfigurátorů jsem nabral inspiraci a udělal jsem si hrubý návrh mé aplikace.



Obrázek 2.3: Kofigurátor Mercedes-Benz.

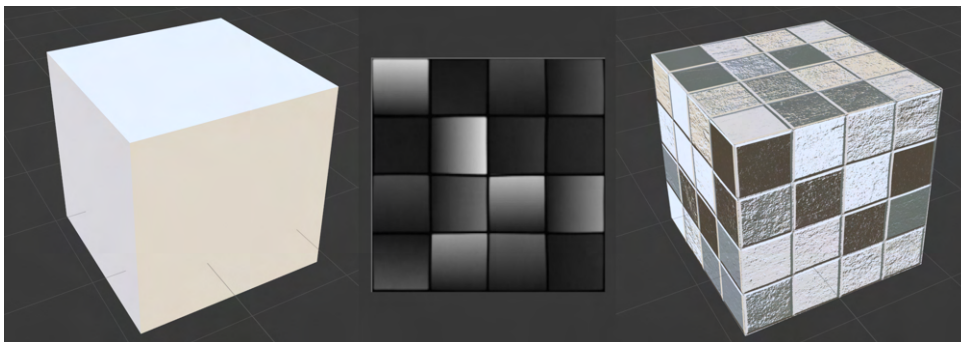
2.2 Simulace a zobrazování vodní hladiny

Po získání inspirace pro aplikaci jsem potřeboval vyřešit problém samotného bazénu. Zaměřil jsem se tedy na problematiku zobrazování vodní hladiny na počítači.

Vodní hladina se dá na počítači simulovat z rovny plochy pomocí bump mappingu. Bump mapping je metoda, která pomocí funkce či displacement mapy dokáže vytvořit vizuálně zvlněný (nebo jinak deformovaný) povrch. Tento efekt je poměrně přesvědčivý pro menší úpravy povrchu, nebo je-li objekt pozorován z dálky. Když se koukneme například na kouli, zvlněnou bump mappingem, tak se uprostřed bude zdát doopravdy zvlněná, ale na okrajích bude stále vidět její hladký povrch. Bump mapping můžeme popsat následující funkcí [1]:

$$p' = p + d(p)n(p). \quad (2.1)$$

Zde p' představuje výsledný přemístěný bod, $d(p)$ je offset bodu získaný z displacement textury pro bod p a $n(p)$ značí normálu pro bod p .

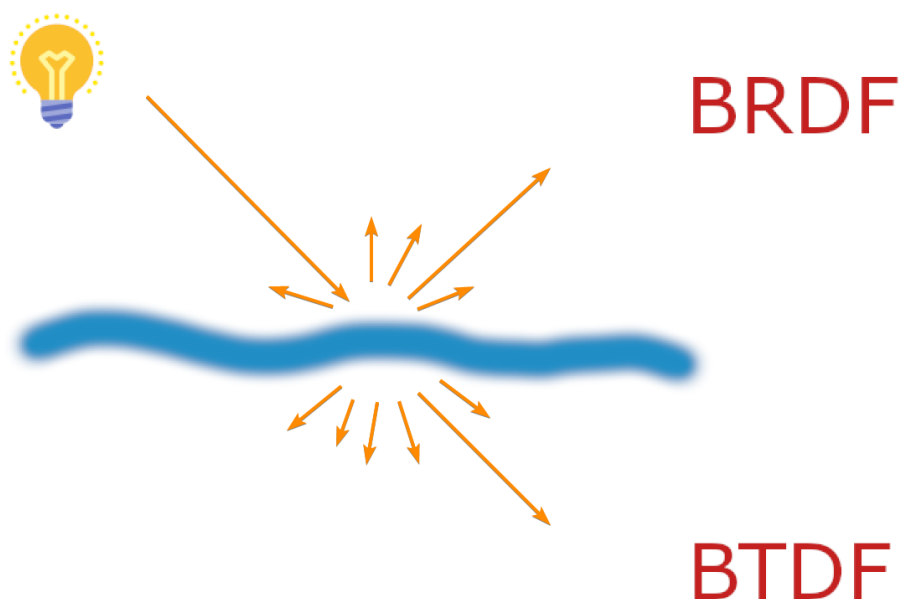


Obrázek 2.4: Aplikace displacement mapy v Blenderu

Jelikož se na vodní hladinu bazénu budeme dívat ze shora, a ne přímo ze strany, tak je bump mapping ideální pro vytvoření a animaci vodní hladiny v bazénu.

Při zobrazování vodní hladiny je důležité vědět, kolik světla se odrazí od povrchu a kolik projde skrz. To popisuje hned několik funkcí, některé jsou realističtější, jiné jsou méně náročné na výpočet. Popíší dvě z nich: obousměrná distribuční funkce odrazu světla (BRDF - Bidirectional Reflectance Distribution Function) a obousměrná distribuční funkce propustnosti světla (BTDF - Bidirectional Transmittance Distribution Function), které když se spojí dohromady nám dají obousměrná distribuční funkce rozptylu světla (BSDF - Bidirectional Scattering Distribution Function).

BRDF popisuje jak a kam se světlo odrazí při dopadu na povrch. BTDF naopak popisuje kam světlo putuje po průchodu povrchem 2.5.



Obrázek 2.5: Ilustrace BSDF. Ve vrchní polovině obrázku je odraz světla popsán BRDF, ve spodní polovině naopak BTDF.

BRDF je popsána jako

$$f_r(x, \omega_i, \omega_o) = \frac{dL_o(x, \omega_o)}{L_i(x, \omega_i) \cos \theta d\omega_i}, \quad (2.2)$$

kde dL_o je odražená diferenciální zář, dL_i zář vracející se ke směru světla, θ je sklon světla od normály, x značí bod na povrchu, kam dopadá světlo přicházející ze směru světla ω_i , ω_o je směr ke kameře (tedy pozorovateli) [2].

Abychom věděli, kolik světla projde skrz vodní hladinu, je třeba znát i index refraxe (nebo také index lomu vlnění) a Fresnelovi rovnice. Index refrakce značí, o kolik pomaleji se světlo pohybuje v daném prostředí než ve vakuu [1]. Index refrakce η můžeme vypočítat pomocí následující rovnice, kde c značí rychlost světla ve vakuu a v značí rychlost světla v daném prostředí:

$$\eta = \frac{c}{v}. \quad (2.3)$$

Fresnelovi rovnice popisují, kolik světla se od povrchu materiálu odrazí. Vycházejí ze zákona odrazu vlnění (úhel odrazu se rovná úhlu dopadu):

$$\theta_o = \theta_d, \quad (2.4)$$

a Snellova zákona:

$$\eta_1 \sin \theta_1 = \eta_2 \sin \theta_2. \quad (2.5)$$

Pro výpočet Fresnelovi odrazivosti potřebujeme znát indexy refrakce obou materiálů:

$$\begin{aligned} r_{\parallel} &= \frac{\eta_t \cos \theta_i - \eta_i \cos \theta_t}{\eta_t \cos \theta_i + \eta_i \cos \theta_t}, \\ r_{\perp} &= \frac{\eta_i \cos \theta_i - \eta_t \cos \theta_t}{\eta_i \cos \theta_i + \eta_t \cos \theta_t}. \end{aligned} \quad (2.6)$$

Zde r_{\parallel} a r_{\perp} značí Fresnelovu odrazivost pro paralelní a kolmé polarizované světlo. Pro nepolarizované světlo můžeme Fresnelovu odrazivost vypočítat jako [1]:

$$F_r = \frac{1}{2}(r_{\parallel}^2 + r_{\perp}^2). \quad (2.7)$$

Tabulka 2.1 níže uvádí příklady indexů refrakce různých materiálů [1].

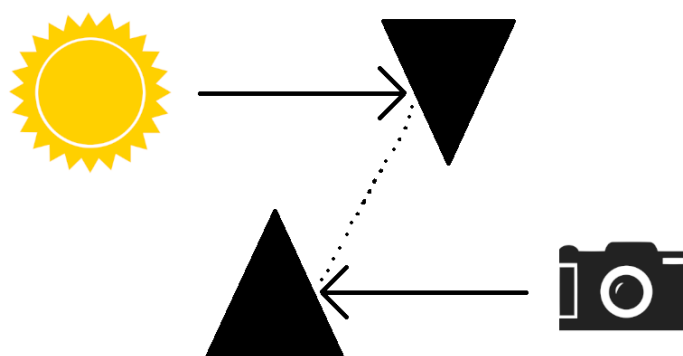
Materiál	Index refrakce
Vakuum	1.0
Vzduch	1.00029
Led	1.31
Voda (20°C)	1.333
Sklo	1.5-1.6
Diamant	2.42

Tabulka 2.1: Tabulka indexů refrakce různých materiálů

Dalším problémem při zobrazování vody je takzvaný objemový, či volumetrický, rozptyl (volumetric scattering). Světlo procházející prostředím ovlivňuje v zásadě tři faktory [1]: absorpce (absorption), emise (emission) a rozptýlení (scattering). Absorption ovlivňuje, jak moc je světlo tlumeno. Krásným příkladem je světlo v mlze. Emission je přesný opak - světlo je zesíleno při průchodu materiálem. Může to být z důvodu nějaké chemické či termální reakce. Nejzajímavější je scattering: při průchodu prostředím některé paprsky světla narazí do malých částecek a odrazí se jiným směrem. To zapříčiní například takzvané měkké stíny.

Obousměrný (bidirectional) path tracing, narozdíl od jednosměrného (unidirectional), kde sledujeme paprsky vedoucí od kamery, sleduje paprsky, které vedou jak od kamery do scény, tak od zdroje světla. Díky tomu dokáže lépe zobrazovat složitější osvětlení a navíc narozdíl od progresivního fotonového mapování nerozmazává osvětlení scény [1].

Další způsob výpočtu osvětlení, vhodný pro tento projekt, je metropolitní světelný přenos (Metropolis Light Transport - MLT). Tato metoda je založena



Obrázek 2.6: Obousměrný path tracing, paprsky vychází ze zdroje světla a od kamery zároveň

na principu na Metropolis-Hastings algoritmu, který vychází z Monte Carlo integrování pomocí Markovova řetězce (Markov Chain Monte Carlo), kdežto předešlé metody výpočtu osvětlení vycházejí pouze z Monte Carlo integrování (Monte Carlo integration). Metropolitní světelný transport vytváří cesty mezi kamerou a světelným zdrojem stejně jako bidirectional path tracing, ale s tím rozdílem, že každá nová cestá je mírně upravená předešlá. MLT upravuje tyto cesty podle pěti hlavních pravidel, z čehož tři jsou zaměřeny právě na složité případy - jako jsou například kaustiky [1].

Po zmapování různých metod zobrazování vody na počítači jsem mohl začít zkoumat praktickou implementaci.

Kapitola 3

Výběr rendereru

3.1 Srovnání

Nejprve mě program napadlo vytvořit v nějakém game enginu, konkrétně Unity či Unreal Engine. Výhodou by, díky okamžitému renderingu v reálném čase, byla mnohem větší přizpůsobitelnost. Uživatel by hned viděl, jak se vzhled bazénu mění s časem nebo počasím. Dokonce by si i mohl nahrát vlastní textury. Na druhou stranu velkou nevýhodou je nerealističnost. Z toho důvodu jsem se rozhodl vytvořit si krátké animace pomocí nějakého rendereru a samotný program mezi nimi bude pouze přepínat v závislosti na vybraných parametrech.

Rozhodoval jsem se především mezi těmito dvěma 3D modelářskými programy: Maya a Blender.

Maya je komerční program firmy Autodesk. Pro rendering scén používá fotorealistický Arnold Renderer, jenž je založen na metodě sledování cest (path tracing) a paprsků (ray tracing) za pomoci Monte Carlo integrování.

Blender je open source 3D modelovací program, který pro realistické renderování využívá Cycles Renderer. Ten je, stejně jako Arnold Renderer, vykresluje scénu Monte Carlo path tracingem [3] [4].

3.2 Blender a LuxCoreRenderer

Pro vytvoření mé scény jsem si zvolil Blender. A to ze dvou prostých důvodů: je zdarma a již jsem v něm pracoval.

Bohužel, jak jsem v průběhu své práce zjistil, Cycles renderer (potažmo i Arnold Renderer) nedokáže pořádně zobrazovat kaustiky [3] [4]. Důvodem je, že používají pouze jednosměrný path tracing (unidirectional path tracing), takže výsledné kaustiky jsou velmi jemné a ne ostré, jak jsme zvyklí z reality. Na realistické vypočítání a zobrazování kaustik je třeba využít jednu z obousměrných metod, například bidirectional path tracing či metropolis light transport.

Začal jsem tedy hledat alternativní renderer, který bych mohl jako doplněk nainstalovat do Blenderu a následně použít pro renderování mé scény. Tři open source renderery, které mě zaujaly nejvíce jsou: LuxCoreRenderer, appleseed

a YafaRay. LuxcoreRenderer vypočítává osvětlení scény obousměrným path tracingem, který kombinuje s Metropolis samplingem. Zkusil jsem tedy svou scénu vyrenderovat LuxCoreRenderem s vynikajícími výsledky - kaustiky byly, narozdíl od Cycles rendereru, krásně vidět. Viz obrázek 5.2b nebo ??.

V průběhu práce na tomto projektu vyšla nová verze Blenderu, která umožnila renderování kaustik pomocí Cycles rendereru. Po porovnání Luxcore a Cycles rendererů jsem se ve finální aplikaci vrátil k Cycles rendereru, především kvůli nestabilní a pouze experimentální verzi LuxCore rendereru pro Blender.

Kapitola 4

Modelování bazénu

4.1 Základní tvar

Nejprve jsem vytvořil velice jednoduchý bazén ve tvaru dutého kvádrů zapuštěného v zemi. Ten jsem několikrát zduplikoval a aplikoval na ně různé volně použitelné textury pod licenci CC0 [5]. Scénu jsem osvětlil jak směrovým světlem představující slunce, tak různými HDRI mapami (vybral jsem, stejně jako u textur, volně dostupné pod licenci CC0 [7]). Jako vodní hladinu jsem použil pouze rovinu, jejíž normály jsem upravil pomocí bump mappingu, abych vytvořil umělou hladinu vody. Následně jsem jí zanimal, ale výsledek vypadá velice uměle. Viz obrázek 5.1.

Proto jsem zkusil vytvořit vodu jiným způsobem. Vodní hladinu již nepředstavovala pouze rovina, ale kvádr. Jeho geometrii jsem rozdělil na mnoho vertikálních hranolů a aplikoval jsem ocean modifier. Ten dokáže poměrně realisticky simulovat vodní hladinu. Dále jsem se pokusil vytvořit materiál, který by dokázal simulovat, či alespoň napodobovat, kaustiky. Toho jsem se pokusil docílit promítáním zostřených stínů hladiny na dno bazénu. Výsledek byl ale málo realistický, protože tyto umělé kaustiky byly moc tlusté, nedostatečně ostré a v animaci se nechovaly tak, jak by člověk čekal.

Pokračoval jsem vylepšením modelu bazénu. Nový model má tvar hrušky, vedou do něj schody, vytvořené ze stejného materiálu, a na stěnách má několik světel. Chvilí jsem experimentoval s jejich tvarem. Začal jsem dlouhými ovály, které byly částečně umístěny nad hladinou. Nakonec jsem světlo vymodeloval ze zploštělého elipsoidu s kovovým rámem. Tyto světla jsem umístil zhruba do poloviny hloubky bazénu, zcela ponořené pod vodní hladinou.

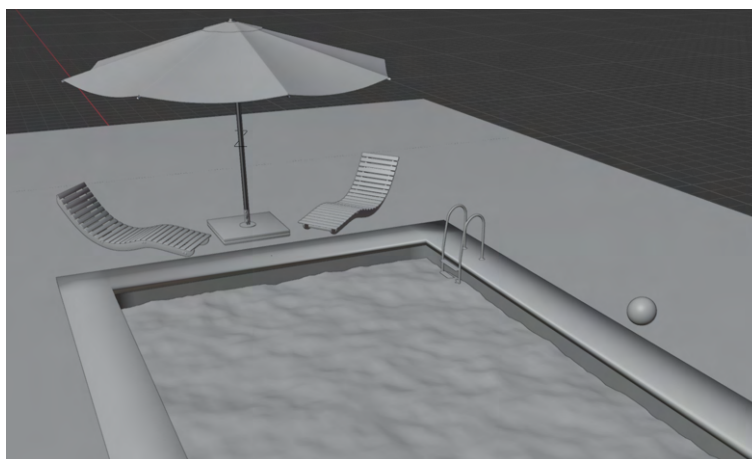
Jelikož jsem nebyl spokojen s výsledkem umělých kaustik, začal jsem hledat alternativní renderer, kterým bych nahradil Cycles. Zvolil jsem LuxCoreRenderer, jak jsem zmínil v předešlé kapitole. S jeho pomocí jsem již dokázal zobrazit požadované kaustiky. Bohužel jsem krátce poté změnil hardware a LuxCoreRenderer mi přestal fungovat.

Mezitím vyšla nová verze Blenderu, která začala podporovat vykreslování kaustik. Bazén jsem znovu předělal, zpět na jednoduchý obdélníkový tvar, jelikož u zaoblených tvarů se deformovala textura dlaždic. Okolo bazénu jsem dal jednoduchou betonovou dlažbu. Původně jsem chtěl trávník, ale zjistil jsem, že je to nad mé momentální schopnosti, tak jsem to odložil na později.

Do okolí jsem umístil několik volně dostupných objektů pod CC0 licenci, abych scénu trochu oživil. Pro bazén jsem vytvořil dřevěný rám a vymodeloval schůdky se zábradlím.

HDRI mapy jsem musel opustit, jelikož Cycles renderer dokáže simulovat kaustiky pouze z přímého zdroje světla. Zaměřil jsem se především na dvě počasí: slunečno a zataženo. Slunečné počasí jsem nasimuloval pomocí jednoduchého zdroje světla nastaveného jako "sun", což jsem nemohl udělat u zataženého počasí. Když jsem zkusil pouze ztlumit slunce, tak jsem pořád dostával velice ostré stíny od objektů, což není moc realistické. Proto jsem zdroj světla nastavil jako "area light", a jej umístil daleko od scény. Tím jsem získal měkké stíny, které jsou více realistické pro zatažené počasí s nižší viditelností. Nakonec jsem přidal i noc: scénu jsem osvětlil velmi slabým sluncem a do bazénu jsem přidal několik světel.

Voda se skládá z 12288 polygonů, bazén z 3766, lem bazénu z 22 a okolní betonové dlaždice pouze z 12. Stažené modely mají 109653 polygonů.



Obrázek 4.1: Scéna v Blenderu, bez textur

4.2 Jednotlivé konfigurace

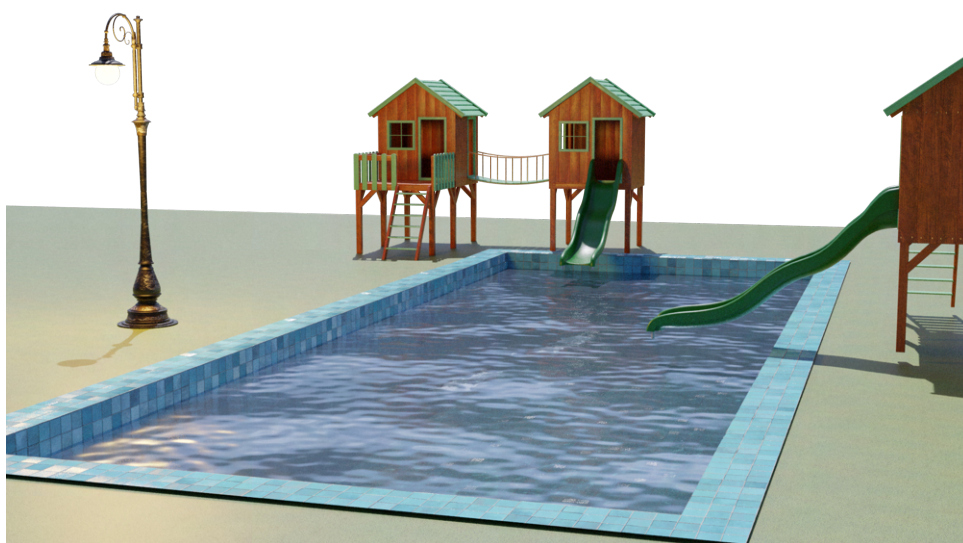
Ve finální aplikaci si uživatel může vybrat různé konfigurace a aplikace mu ukáže předrenderovaný obrázek. První na výběr materiál bazénu: tyrkysová mozaika, zelenomodré dlaždice nebo modrobílý vzor. Dále je samotný model bazénu: jeden má schůdky se zábradlím a druhý má pouze nerezový žebřík. Jako další si může vybrat, jestli má bazén mít dřevěný lem. Dekorace kolem bazénu původně měly jít vypnout a zapnout, ale to jsem nakonec zavrhl z důvodů jak časové, tak paměťové úspory. Dále počasí: lze vybrat buď slunečné, zatažené nebo noc. V noci je bazén osvětlen světly umístěnými na stěnách bazénu. Nakonec uživatel vybírá pohled na bazén. Ten je možné zvolit přímo zepředu nebo z boku.

Kapitola 5

Rendering scény

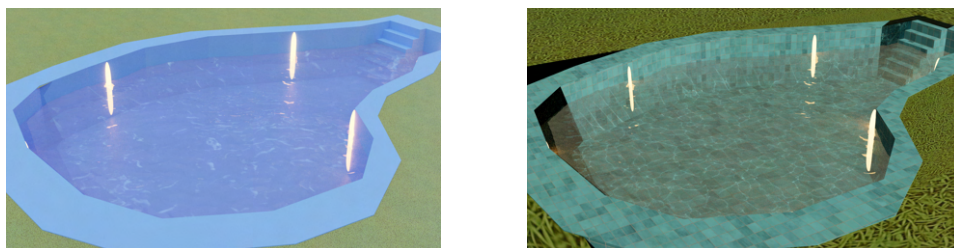
5.1 Vykreslování různých konfigurací

Jeden z prvních vyrenderovaných snímků bazénu představuje obrázek 5.1, který byl renderován přes Cycles na 128 samplů. Zkoušel jsem do scény umístit nějaké volně dostupné modely kolem bazénu, abych viděl odrazy ve vodě.

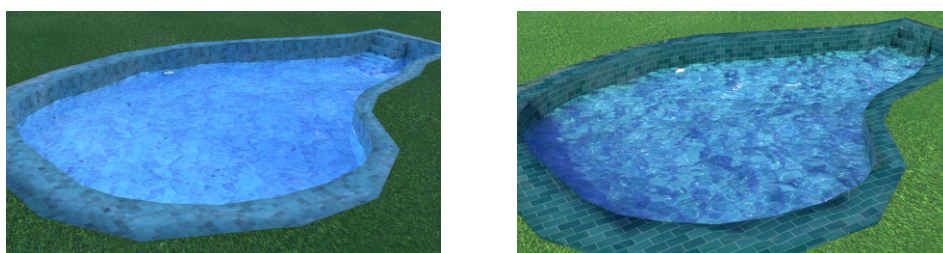


Obrázek 5.1: Jednoduchý bazén, Cycles

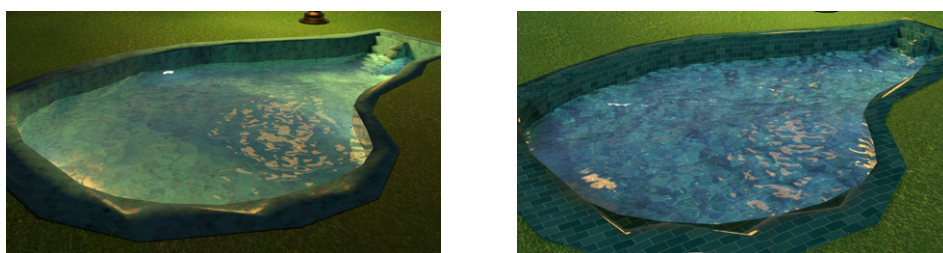
V dalších snímcích jsem změnil tvar bazénu a přidal světla dovnitř. Obrázky 5.2 renderován také pomocí Cycles na 128 samplů. Poté jsem přepnul na LuxCoreRenderer, který jsem nechal běžet zhruba 10-15 minut na 128 samplů. Všechny tyto rendery jsou ve FullHD (1920 × 1080) rozlišení a proběhly bez denoiseru, takže v nich lze vidět zrnité oblasti.



Obrázek 5.2: Cycles a LuxCore, složitější model.



Obrázek 5.3: Cycles den, 512 Samples - 23 min vlevo, 32 min vpravo



Obrázek 5.4: Cycles noc, 512 Samples - 28 min vlevo, 31 min vpravo.

Bohužel mi po změně hardwaru z procesoru Intel 5500K a grafické karty GTX1060 na procesor AMD RYZEN 5 5600X a grafickou kartu RTX3070 přestal fungovat LuxCoreRenderer, tudíž následující renderování ?? jsou pouze pomocí Cycles, když ještě nepodporoval kaustiky. Všechny z následujících snímků byly renderovány ve 4K (3840 × 2160) rozlišení na 512 samples a měly zapnutý denoiser, který výsledný obrázek zbavil šumu. Renderování trvalo mezi 23 až 32 minutami.

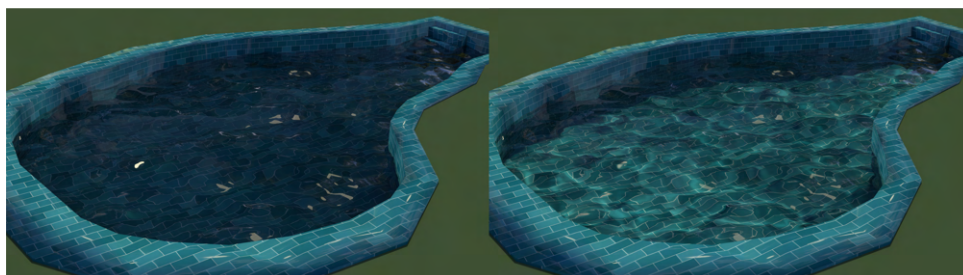
Dále jsem vyrenderoval dvě animace: jednu pro bazén ve dne a jednu pro bazén v noci. Tyto animace byly renderovány na 128 vzorků, pouze ve FullHD rozlišení a mají 100 snímků. Každý snímek trval vyrenderovat okolo 2.5 minut. Celá animace trvala krátce pod 5 hodin. Animace a ostatní snímky jsou přiložené v zazipovaném archivu.

Po znovuzprovoznění LuxCoreRenderera jsem vymodeloval klasický obdélníkový bazén s dřevěným lemem. Render jsem nechal běžet 42 minut, maximum světelných cest jsem nastavil na 32 a celkově proběhlo 7450 vzorků 5.5.



Obrázek 5.5: LuxCore renderer - 42 min.

Před opuštěním hruškovitého tvaru bazénu (ve prospěch jednoduchého obdélníku) jsem vyrenderoval kaustiky pomocí Cycles Rendereru (rozlišení 4K, 256 vzorků). Viz obrázek 5.6.



Obrázek 5.6: Kaustiky v Cycles rendereru.

Níže je několik vyrenderovaných obrázků finálního bazénu pro některé vybrané konfigurace. Všechny obrázky byly renderovány pomocí Cycles rendereru v rozlišení 4K, čili 3840 na 2160 pixelů. Počet vzorků (samples) jsem nastavil na 256. Snímky byly renderovány v nové verzi Blenderu, kde Cycles renderer podporuje vykreslování kaustik.

Použitý hardware: NVIDIA RTX 3070, AMD RYZEN 5 5600X.

Nastavení v Blenderu jsem ponechal výchozí s výjimkou nastavení "Light Paths", kde jsem zvedl některé hodnoty v kategorii "Max Bounces":

Total=16, Diffuse=4, Transmission=12, Volume=4, Transparent=8.

Časy jednotlivých renderů jsou uvedeny v tabulkách níže 5.2, 5.3, 5.4, 5.5 a 5.6.

Nakonec jsem vyrenderoval tři animaci bazénu, každá s jiným materiálem a v jiném počasí. Všechny animace jsou ve FullHD rozlišení (1920x1080) a mají 60 snímků, ostatní nastavení je identické s předešlými obrázky. Časy renderů jsou uvedeny v tabulce 5.1.

Pro finální aplikaci jsem vytvořil obrázky 54 různých konfigurací a 3 animace. Celkový čas renderování obrázků do aplikace byl 221 minut a 33 sekund. Animace se renderovaly celkem 169 minut a 10 sekund.

5.2 Tabulky a výsledné rendery

Časové údaje v tabulkách jsou ve formátu minuty:sekundy.

Pohled z boku	Průměrný čas jednoho snímku	Celkový čas animace
Tyrkysová mozaika, slunečno	0:50.1	50:05
Modrozelené dlaždice, noc	0:49.8	49:48
Modrobílý vzor, zataženo	1:09.3	69:17

Tabulka 5.1: Časy renderů pro animace finálního bazénu.

Slunečno, pohled z boku	Schůdky, dřevěný rám	Žebřík, dřevěný rám	Schůdky, žádný rám	Žebřík, žádný rám
Tyrkysová mozaika	03:19	03:30	03:51	03:31
Modrozelené dlaždice	01:51	01:58	01:44	01:49
Modrobílý vzor	02:53	03:10	03:08	03:58

Tabulka 5.2: Časy renderů pro různé konfigurace při slunečném počasí.



Obrázek 5.7: Tyrkysová mozaika, slunečný den

Zataženo, pohled z boku	Schůdky, dřevěný rám	Žebřík, dřevěný rám	Schůdky, žádný rám	Žebřík, žádný rám
Tyrkysová mozaika	04:54	08:31	04:25	06:53
Modrozelené dlaždice	02:20	02:31	02:22	02:29
Modrobílý vzor	03:49	06:46	03:40	04:06

Tabulka 5.3: Časy renderů pro různé konfigurace při slunečném počasí.

Slunečno, pohled zepředu	Schůdky, dřevěný rám	Žebřík, dřevěný rám	Schůdky, žádný rám	Žebřík, žádný rám
Tyrkysová mozaika	04:34	04:33	04:05	04:37
Modrozelené dlaždice	02:44	02:49	02:34	02:28
Modrobílý vzor	03:48	03:49	04:11	04:10

Tabulka 5.4: Časy renderů pro různé konfigurace za horší viditelnosti.



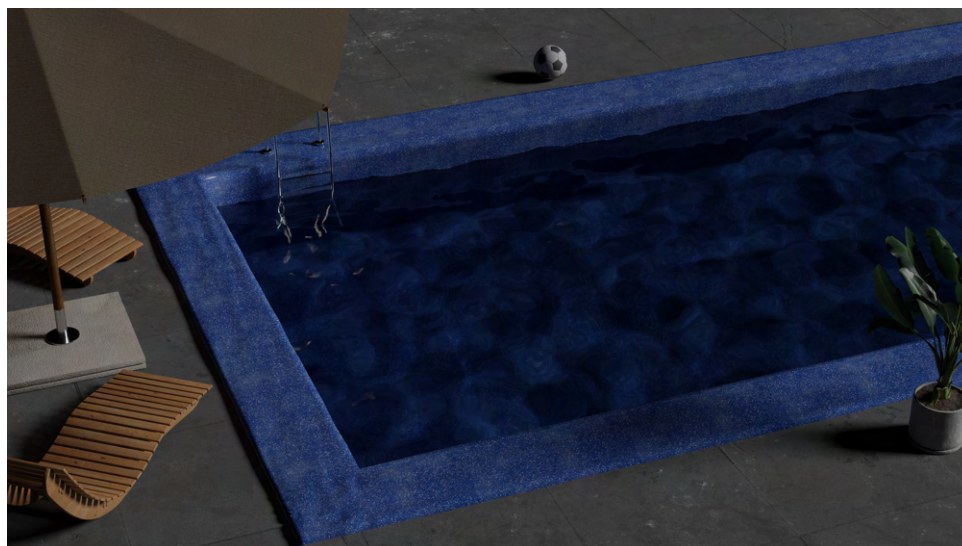
Obrázek 5.8: Modrozelené kachličky, slunečný den

Zataženo, pohled zepředu	Schůdky, dřevěný rám	Žebřík, dřevěný rám	Schůdky, žádný rám	Žebřík, žádný rám
Tyrkysová mozaika	05:33	05:56	05:25	06:25
Modrozelené dlaždice	03:32	03:59	03:11	03:26
Modrobílý vzor	05:12	05:07	05:01	05:21

Tabulka 5.5: Časy renderů pro různé konfigurace za horší viditelnosti.

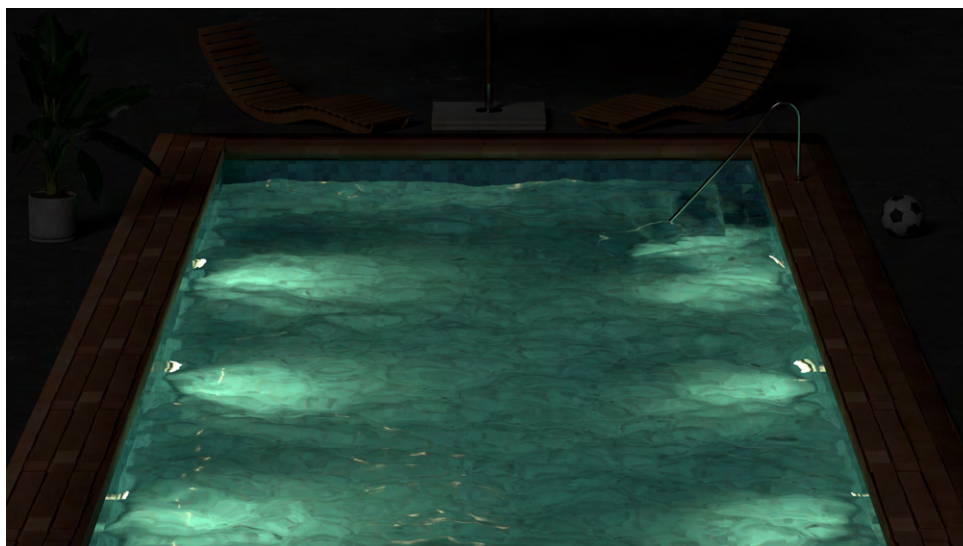
Noc, světla v bazénu	Pohled z boku	Pohled ze předu
Tyrkysová mozaika	05:19	6:39
Modrozelené dlaždice	03:42	4:55
Modrobílý vzor	4:49	6:11

Tabulka 5.6: Časy renderů pro různé konfigurace v noci.



Obrázek 5.9: Modrobílý vzor, nižší viditelnost

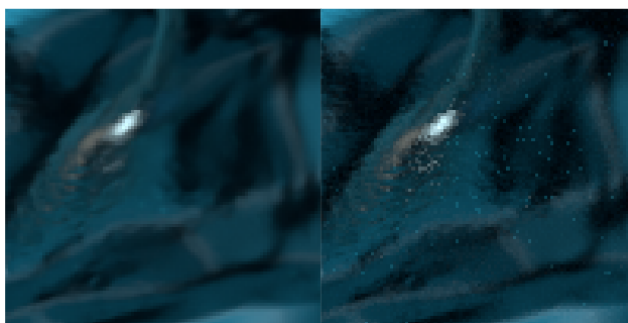
Obrázek 5.10: Tyrkysová mozaika, noční osvětlení



Obrázek 5.11: Modrobílé kachličky, noční osvětlení

5.3 Srovnání Cycles a LuxCore rendererů

Do Blenderu jsem nainstaloval experimentální verzi LuxCoreRendereru (stabilní verze byla pouze pro starší verzi Blenderu, který nepodporuje kaustiky) a upravil jsem materiál vody pro LuxCore. Následně jsem vyrenderoval dva snímky pomocí Cycles - u druhého jsem vypnul denoiser (256 Samples, rozlišení 4K, stejné nastavení jako předchozí). Jediné místo s šumem, které jsem na obrázku našel, bylo kolem kovové tyče ve vodě u schodů 5.12. LuxCore renderer jsem nastavil co nejpodobněji jako Cycles: 16 Total Light Paths, 4 Diffuse, 4 Glossy, 6 Specular. Počet fotonů v Photon GI Cache jsem snížil z 20 milionů na 10. Jelikož výchozí nastavení LuxCore rendereru je vykreslovat, dokud ho uživatel nezastaví, nastavil jsem maximální počet vzorků na 256. Vyrenderoval jsem dva obrázky, jeden měl vypnutý denoiser, stejně jako u Cycles, viz obrázky. Jako další jsem nastavil Samples na 512 - jak v Cycles, tak v LuxCore 5.13. Tentokrát jsem zkusil použít bidirectional path tracing (místo obyčejného path tracingu), jediný problém je, že lze vykreslovat pouze pomocí CPU, čímž se několikanásobně prodloužila doba vykreslování 5.7.

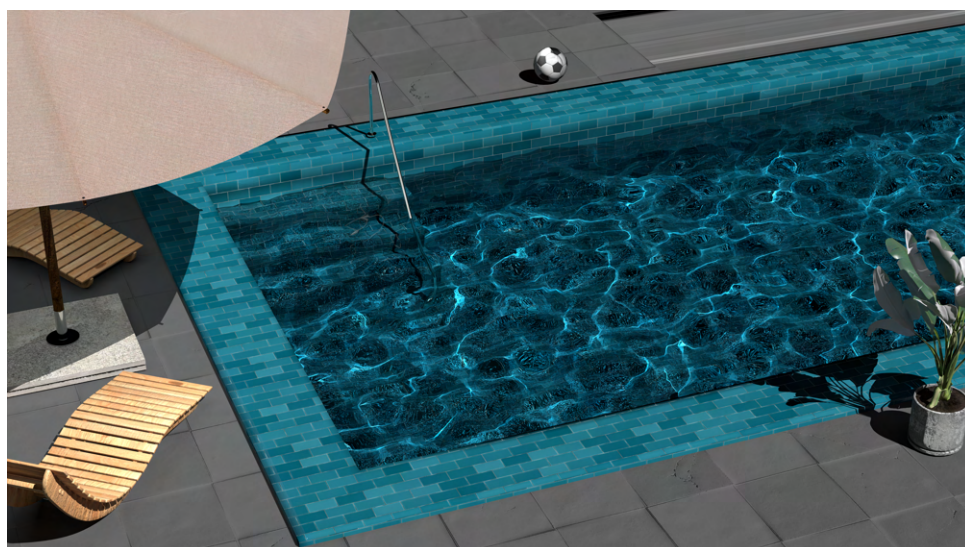


Obrázek 5.12: Denoiser v Cycles - 256 vzorků.

	Cycles	LuxCore
256 vzorků	2:15	11:28
512 vzorků	2:29	92:27

Tabulka 5.7: Časy renderů Cycles a LuxCore. LuxCore na 512 vzorků byl renderován pouze pomocí CPU.

LuxCore vykresluje mnohem lepší kaustiky, ale renderuje mnohem déle. Nejlepších výsledků jsem dosáhl při za použití bidirectional path tracingu, ovšem za cenu zhruba čtyřnásobné doby renderu. 512 vzorků nebylo dostatek, aby se voda zbavila šumu. Samotná textura vody se mi v experimentální verzi LuxCore nepodařila nastavit příliš realisticky. LuxCore používá jiné uzly materiálů (material nodes), takže je potřeba předělat většinu materiálů. Proto jsem radši přešel zpět na Cycles, i když má horší kaustiky. Materiál okolí se nastavuje mnohem snáze, dobrá vykreslování je mnohonásobně nižší, což se nejvíc projeví na době renderování animací. Krom toho se mi během renderu několikrát zasekl Blender a musel jsem znovu nastavovat LuxCore.



Obrázek 5.13: LuxCore, 512 samples, bidirectional path tracing, denoised.

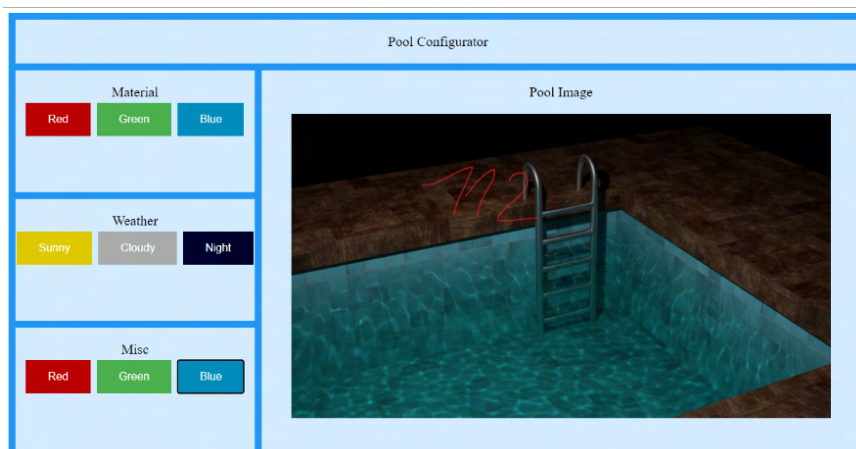
Kapitola 6

Webová aplikace

6.1 Tvorba aplikace

Rozhodl jsem se vytvořit jednoduchou webovou aplikaci, jelikož ta není závislá na operačním systému a uživatel ji může používat i na mobilu. Další z výhod je, že není potřeba nic instalovat ani stahovat. Webovou stránku jsem napsal v jazyce HTML s několika JavaScript funkcemi.

Při tvorbě uživatelského rozhraní jsem se inspiroval v konfigurátorech aut, jak jsem zmiňoval výše. Většinu obrazovky zabírá předrenderovaný obrázek bazénu. Po levé straně je sloupek, kde si uživatel může navolit materiál, doplňky, osvětlení a pohled na bazén.

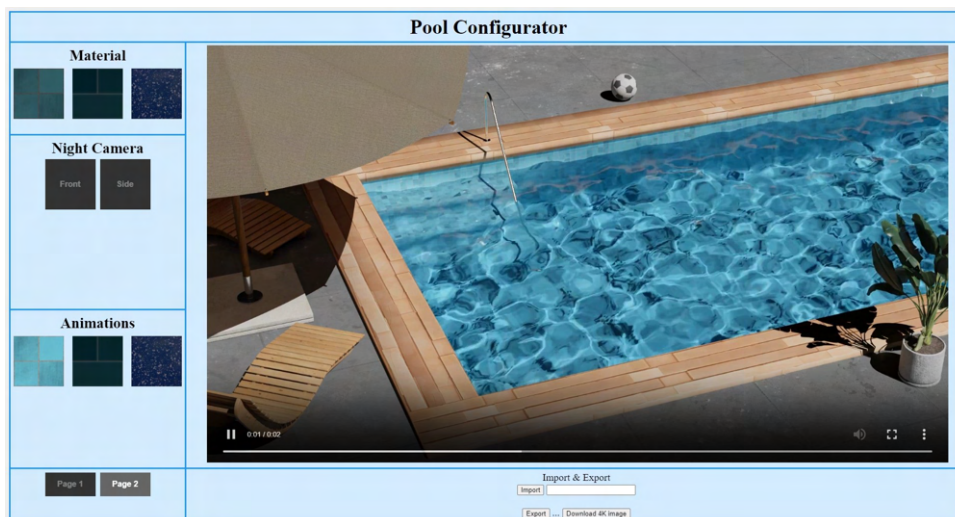


Obrázek 6.1: Raná verze aplikace.

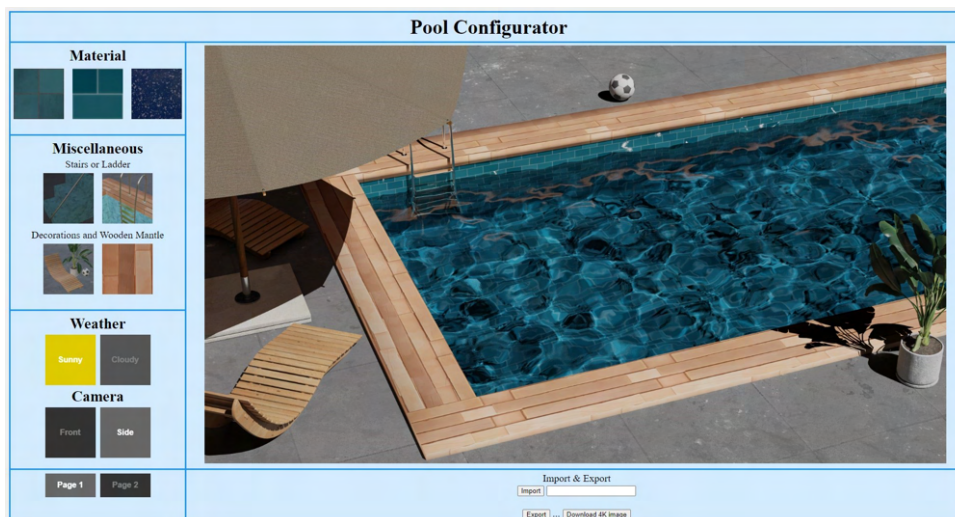
Tlačítka na konfiguraci jsem později vyměnil za interaktivní obrázky, čímž jsem se přiblížil mé inspiraci. Stejně jako u některých konfigurátorů aut jsem přidal možnost uložit si navolenou konfiguraci na později. Pomoci JavaScriptu vygeneruji kód, který se dá kdykoliv vložit do textového pole a tím načíst předchozí hodnoty. Jelikož sloupek začínal být moc plný, vytvořil jsem mu druhou stránku, ze které je možné přehrát několik animací a vybrat více předrenderovaných obrázků. Tlačítka na uložení konfigurace jsem přemístil pod obrázek bazénu abych efektivněji využil prostor.

6.2 Finální vzhled aplikace

Zůstal jsem u jednoduchého stylu, na levé straně je sloupek s tlačítky, u něhož lze přepínat dvě stránky. Největší prostor zabírá obrázek bazénu (nebo animace). Pod ním lze navolenou konfiguraci exportovat nebo naopak importovat (6.2, 6.3).



Obrázek 6.2: Finální vzhled aplikace.



<https://www.overleaf.com/project/5ffc4393b6e0f25d5cf20acd>

Obrázek 6.3: Finální vzhled aplikace.

Kapitola 7

Závěr

V této bakalářské práci jsem uvedl problematiku realistického zobrazování vodní hladiny na počítači, především problém vykreslování kaustik. Také jsem porovnal různé konfigurátory aut, jimiž jsem se inspiroval při tvorbě vlastní aplikace. Dále jsem uvedl svou představu programu pro výběr bazénu. Představil jsem několik 3D modelářských softwarů a rendererů a důvody, proč jsem si pro svou práci vybral zrovna Blender a proč jsem změnil renderer z Cycles na LuxCoreRenderer a následně zpět. Popsal jsem postup vymodelování scény s bazénem a na jaké problémy jsem narazil. Poté jsem představil výsledky vyrenderovaných scén, jak z LuxCoreRendereru, tak z Cycles. Postup tvorby aplikace jsem probral v poslední kapitole.

Konfigurátor by šel rozhodně vylepšit, jak samotná aplikace, tak scéna v Blenderu. Aplikaci by jistě prospěl modernější design a více možností konfigurace. Animace pro každou zvolenou konfiguraci jsem nestihl udělat, protože kombinací je příliš mnoho. Bazén v Blenderu by uvítal rozmanitější okolí (například trávník) a více různých materiálů. Rozhodně bych přidal více pohledů a experimentoval bych s počasím - především déšť je nejzajímavější z technického hlediska.



Literatura

- [1] Matt Pharr, Wenzel Jakob, and Greg Humphreys, *Physically Based Rendering: From Theory To Implementation*, Third Edition, 2018, online, www.pbr-book.org.
- [2] Aleš Koblížek, *Realistické zobrazování vodních ploch*, Diplomová práce, ČVUT FEL, 2020.
- [3] *Blender 4.0 Manual*, 2023, https://docs.blender.org/manual/en/dev/render/cycles/optimizations/reducing_noise.html#path-tracing.
- [4] *Arnold for Maya User Guide*, 2023, <https://docs.arnoldrenderer.com/display/A5AFMUG/Advanced#Advanced-Caustics>.
- [5] *CC0 Textures*, 2023, <https://cc0textures.com/list>.
- [6] *HDRIHaven*, 2023, <https://hdrihaven.com/hdris>.
- [7] *BlenderKit*, 2023, <https://www.blenderkit.com/>.
- [8] *ŠKODA AUTO*, 2023, <https://cc.skoda-auto.com/cze/cs-CZ/>.
- [9] *BMW*, 2023, https://configure.bmw.cz/cs_CZ/configure/.
- [10] *Volkswagen*, 2023, <https://konfigurator.volkswagen.cz/>.
- [11] *Mercedes-Benz*, 2023, <https://www.mercedes-benz.cz/passengercars/configurator.html>.