

## 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: **10.01.2024**

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.  
Leden 2024



## Poděkování

Děkuji mému vedoucímu práce doc. Ing. Jiřímu 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, 9. ledna 2024

## Abstrakt

Tato práce probírá problematiku fotorealistického zobrazování vodní hladiny bazénu. Dále probere různé vykreslovací techniky, vhodné pro zobrazení scény s bazény, a výběr vhodného vykreslovacího nástroje, který využívá jednu z těchto technik. Vykreslené 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 vykreslených snímků, včetně statistik. Dva různé vykreslovací nástroje 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 lighting 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

<b>1 Úvod</b>	<b>1</b>
<b>2 Výběr technologií</b>	<b>3</b>
2.1 Ostatní konfiguratory .....	3
2.2 Simulace a zobrazování vodní hladiny .....	5
<b>3 Výběr rendereru</b>	<b>9</b>
3.1 Srovnání .....	9
3.2 Blender a LuxCoreRenderer .....	10
<b>4 Modelování bazénu</b>	<b>11</b>
4.1 Základní tvar .....	11
<b>5 Rendering scény</b>	<b>13</b>
5.1 Srovnání Cycles Render a LuxCoreRender .....	13
5.2 Vykreslování různých konfigurací	15
5.3 Jednotlivé konfigurace .....	18
<b>6 Webová aplikace</b>	<b>21</b>
6.1 Tvorba aplikace .....	21
6.2 Finální vzhled aplikace .....	22
<b>7 Závěr</b>	<b>25</b>
7.1 Tabulky a výsledné vykreslené obrázky .....	26
<b>Literatura</b>	<b>31</b>

## Obrázky

2.1 Konfigurátor ŠKODA.....	3	7.5 Tyrkysová kachličky, noční osvětlení.....	29
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 popsáný BRDF, ve spodní polovině naopak BTDF. ....	6		
2.6 Obousměrné sledování cest, paprsky vychází ze zdroje světla a od kamery zároveň.....	8		
3.1 Path guiding, podle prezentace Thomase Müllera[13] .....	9		
3.2 Cycles a LuxCore, složitější model. ....	10		
4.1 Scéna v Blenderu, bez textur ...	12		
5.1 Odšumovač v Cycles - 256 vzorků. ....	13		
5.2 LuxCore, 512 samples, bidirectional path tracing, denoised. ....	14		
5.3 Jednoduchý bazén, Cycles .....	15		
5.4 Cycles a LuxCore, složitější model. ....	15		
5.5 LuxCore renderer - 42 min. ....	16		
5.6 Kaustiky v Cycles. ....	16		
5.7 4K rozlišení, 256 vzorků .....	17		
5.8 FullHD, 128 vzorků, polovina Max Bounces .....	17		
5.9 .....	18		
5.10 .....	19		
6.1 Raná verze aplikace. ....	21		
6.2 Finální vzhled konfigurátoru. ...	23		
6.3 Finální vzhled HTA konfigurátoru. ....	24		
7.1 Tyrkysová mozaika, slunečný den	26		
7.2 Modrozelené kachličky, slunečný den .....	27		
7.3 Modrobílý vzor, nižší viditelnost	28		
7.4 Modrozelené kachličky, noční osvětlení.....	29		

## Tabulky

2.1 Tabulka indexů refrakce různých materiálů .....	7
5.1 Časy renderů Cycles a LuxCore. LuxCore na 512 vzorků byl renderován pouze pomocí CPU. . .	14
5.2 Porovnání času vykreslování. . . .	17
7.1 Časy vykreslování pro animace finálního bazénu. ....	26
7.2 Časy vykreslování pro různé konfigurace při slunečném počasí. .	26
7.3 Časy vykreslování pro různé konfigurace při slunečném počasí. .	27
7.4 Časy vykreslování pro různé konfigurace za horší viditelnosti. . .	27
7.5 Časy vykreslování pro různé konfigurace za horší viditelnosti. . .	28
7.6 Časy vykreslování pro různé konfigurace v noci. ....	28





# 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 moci vybrat vzhled bazénu a počasí. Program mu následně ukáže obrázek (či 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 vykreslovací nástroj. Tímto výběrem se zabývám v Kapitole 3. Poté následuje Kapitola 4, kde popisuji svou práci v Blenderu, čili samotné modelování bazénu a volbu textur. Vykreslování scény s bazénem probírám v Kapitole 5. V téže kapitole srovnávám různé vykreslovací nástroje, které jsem použil. V 6. kapitole následuje tvorba vlastní aplikace pro konfiguraci bazénu. Závěrem práce, v poslední kapitole, shrnu svou práci a zamyslím se, jak by se dala má aplikace vylepšit.

Na konci práce jsou tabulky časů vykreslování různých konfigurací bazénů, jejich obrázky a seznam použité literatury.

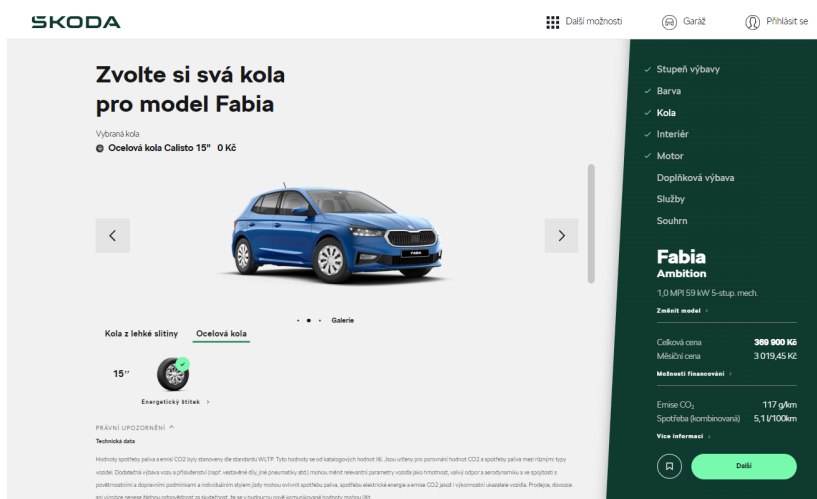


## Kapitola 2

### Výběr technologií

#### 2.1 Ostatní konfigurátory

Abych mohl vytvořit aplikaci, potřeboval jsem najít inspiraci. Tu jsem našel u konfigurátorů 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.



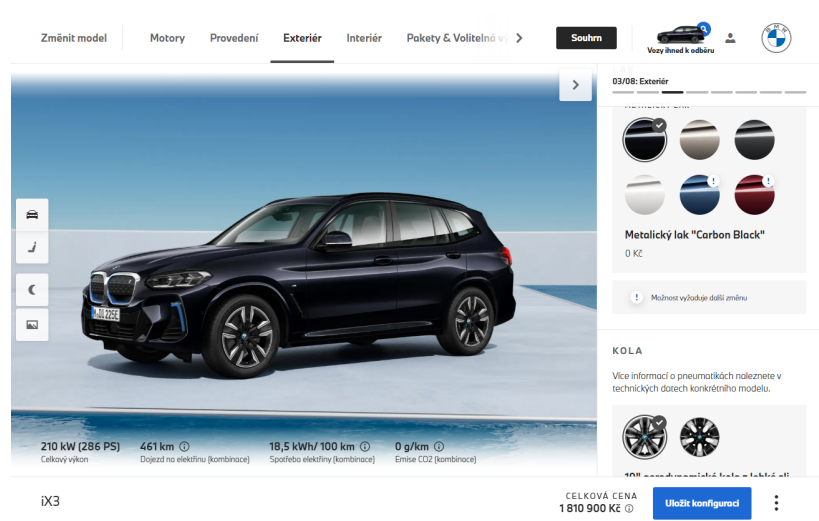
Obrázek 2.1: Konfigurátor ŠKODA.

Nejprve jsem zkoumal konfigurátor ŠKODA [8]. Při změně materiálu a doplňků, se mění 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. Na rozdíl 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

## 2. Výběr technologií

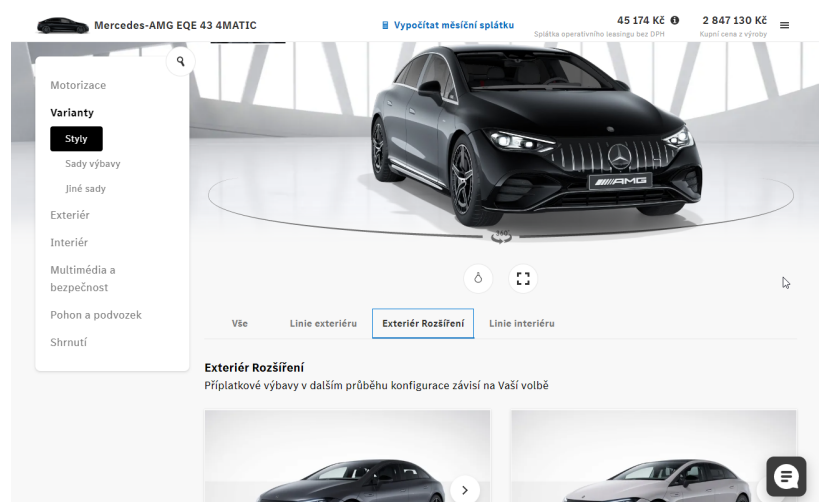
BMW konfigurátoru je možnost uložit si vybranou konfiguraci na později.



Obrázek 2.2: Kofigurátor BMW.

Jako další jsem našel konfigurátor aut Volkswagen [10]. Ten mě velice zklamal, neboť má pouze předvykreslených 12 pohledů na exteriér auta, takže s ním zákazník nemůže 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: některé věci vybírám ve sloupci na straně obrazovku a jiné pod obrázkem auta 2.3.

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



Obrázek 2.3: Kofigurátor Mercedes-Benz.

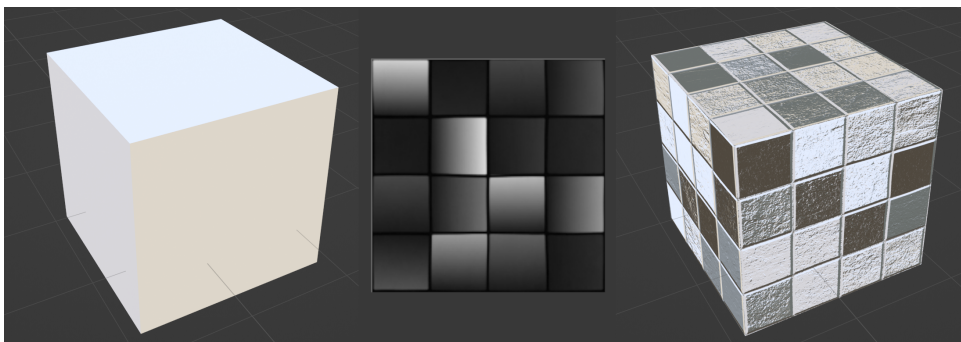
## 2.2 Simulace a zobrazování vodní hladiny

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 metody zvané bump mapping. Tato metoda, která pomocí funkce či mapy posunu (displacement map) 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 například koukneme na takto zvlněnou kouli, tak se uprostřed bude zdát doopravdy zvlněná, ale na okrajích bude stále vidět její hladký povrch. Bump mapping můžeme popsat následující funkcí [1]:

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

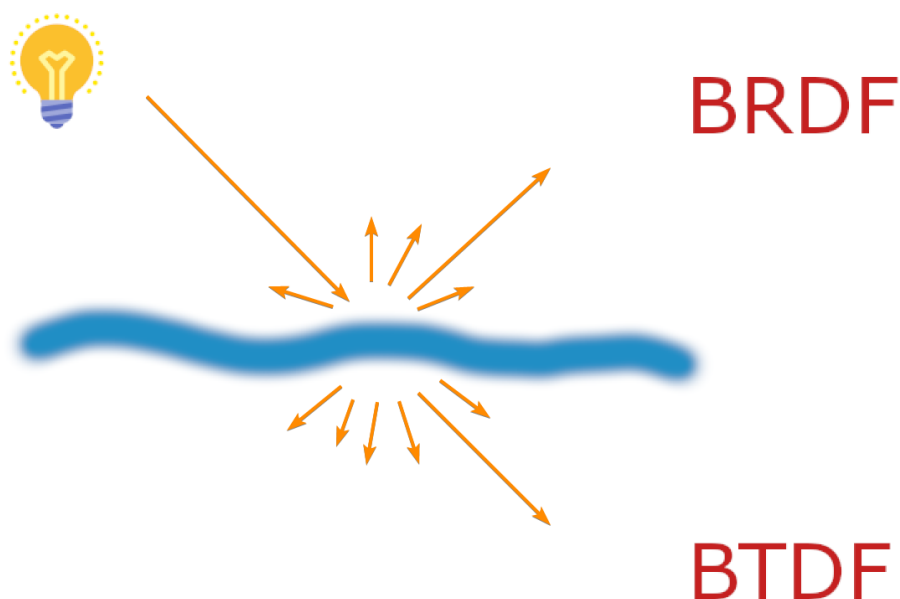
Zde  $p'$  představuje výsledný přemístěný bod,  $d(p)$  je offset bodu získaný z displacement mapy textury pro bod  $p$  a  $n(p)$  značí normálu pro bod  $p$ .



**Obrázek 2.4:** Aplikace displacement mapy v Blenderu

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 popsáný 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,  $\theta$  je sklon světla od normály,  $x$  značí bod na povrchu, kam dopadá světlo přicházející ze směru světla  $\omega_i$ ,  $\omega_o$  je směr ke kameře (tedy pozorovateli) [2].

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

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

Fresnelovy 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 Fresnelovy 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 5.2 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

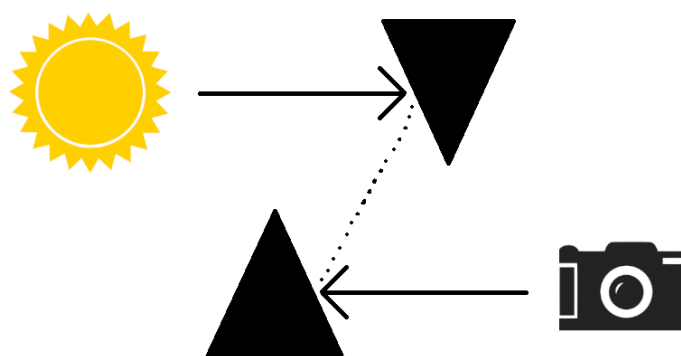
**Tabulka 2.1:** Tabulka indexů refrakce různých materiálů

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

Obousměrné sledování cest (bidirectional path tracing), na rozdíl 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, oproti progresivnímu fotonovému mapování, nerozmazává osvětlení scény [1].

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





**Obrázek 2.6:** Obousměrné sledování cest, paprsky vychází ze zdroje světla a od kamery zároveň

na principu 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). Metropolisův světelný přenos vytváří cesty mezi kamerou a světelným zdrojem, stejně jako obousměrné sledování cest, ale s tím rozdílem, že každá nová cesta 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 napadlo vytvořit program v nějakém herním enginu, konkrétně Unity či Unreal Engine. Výhodou by, díky okamžitému vykreslování 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 obrázky a krátké animace pomocí softwaru pro vytváření 3D grafiky. Můj program bude přepínat mezi vygenerovanými obrázky a animacemi v závislosti na výběru uživatele. Pokud obrázky nebudou pro danou konfiguraci vygenerované, tak se spustí vybraný modelářský software a obrázek vygeneruje.

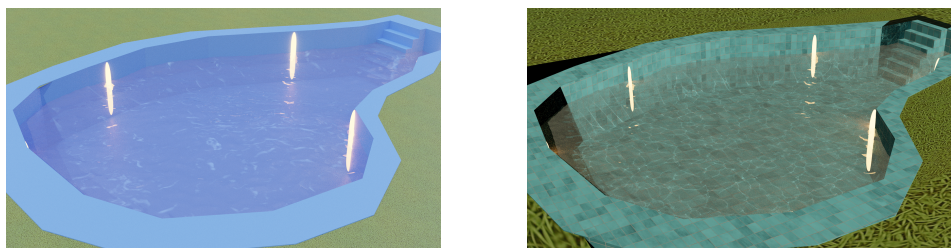
Rozhodoval jsem se především mezi těmito dvěma 3D modelářskými programy: Maya a Blender.

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

Blender je otevřený software (open-source) pro 3D modelování, který pro realistické renderování využívá Cycles Renderer. Ten, stejně jako Arnold Renderer, vykresluje scénu pomocí sledování cest Monte Carlo algoritmem [3] [4]. V jedné z posledních aktualizací bylo do Cycles přidáno vedení cest (path guiding) pomocí Intel® Open Path Guiding Library. Vedení cest mimo jiné pomáhá s vykreslováním jednoduchých kaustik vzniklých po průchodu světla vodní hladinou.



Obrázek 3.1: Path guiding, podle prezentace Thomase Müllera[13]



Obrázek 3.2: Cycles a LuxCore, složitější model.

## 3.2 Blender a LuxCoreRender

Pro vytvoření mé scény jsem si zvolil Blender. A to z několika prostých důvodů: je zdarma, již jsem v něm pracoval a vím, že vykreslení scény se dá spustit z příkazové řádky.

Jelikož vykreslování kaustik v Cycles je poměrně nová funkce, začal jsem hledat alternativní možnost. Donedávna Cycles renderer (potažmo i Arnold Renderer) nedokázal pořádně zobrazovat kaustiky. Důvodem je, že používají pouze jednosměrné sledování cest (unidirectional path tracing), takže výsledné kaustiky jsou velmi jemné a ne ostré, jak známe z reality [3] [4]. Na realistické vypočítání a zobrazování kaustik je třeba využít jednu z obousměrných metod, například obousměrné sledování cest nebo Metropolisův světelný přenos (případně nové vedení cest).

Začal jsem tedy hledat alternativní renderer, který bych mohl nainstalovat do Blenderu jako doplněk a následně jej mohl použít pro vykreslení mé scény. Tři open-source renderery, které mě zaujaly nejvíce jsou: LuxCoreRender, appleseed a YafaRay. LuxcoreRender vypočítává osvětlení scény obousměrným vedením cest, který kombinuje s technikou Metropolis Sampling. Zkusil jsem tedy svou scénu vyrenderovat pomocí LuxCoreRenderu s vynikajícími výsledky - kaustiky byly krásně viditelné. Velký problém byla ovšem dlouhá výpočetní doba. Viz obrázek 3.2.

Po vyzkoušení kaustik v LuxCoreRenderu jsem zkusil vykreslit kaustiky pomocí Cycles. Kaustiky nebyly tak ostré, ale zůstaly krásně viditelné a hlavně se několikanásobně snížila doba vykreslování. Hlavním důvodem je, že ve výchozím nastavení LuxCoreRender vykresluje dokud se manuálně nezastaví (lze nastavit ukončení vykreslování po určité době). Pro stejnou dobu vykreslování byl v LuxCore obrázku viditelně větší šum. Proto jsem usoudil, že Cycles bude ideální renderer pro mou aplikaci.

## 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 licencí CC0 [5]. Scénu jsem osvětlil jak směrovým světlem představující slunce, tak různými HDRI mapami (vybral jsem, stejně jako u textur, volně dostupné pod licencí CC0 [6]). 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 v kapitole 5 5.3.

Proto jsem zkusil vytvořit vodu jiným způsobem. Vodní hladinu stále tvoří plocha, jejíž geometrii jsem rozdělil na mnoho polygonů a následně jsem aplikoval Ocean modifier. Ten vychází z open-source Houdini Ocean Toolkitu, který používá algoritmy Jerryho Tessendorfa pro vykreslování vln oceánu HOT. Tato funkce dokáže poměrně realisticky simulovat vodní hladinu a dá se jednoduše animovat. Pro mé účely jsem použil klidný oceán s nízkými vlnami, abych se co nejvíce přiblížil hladině vody v bazénu.

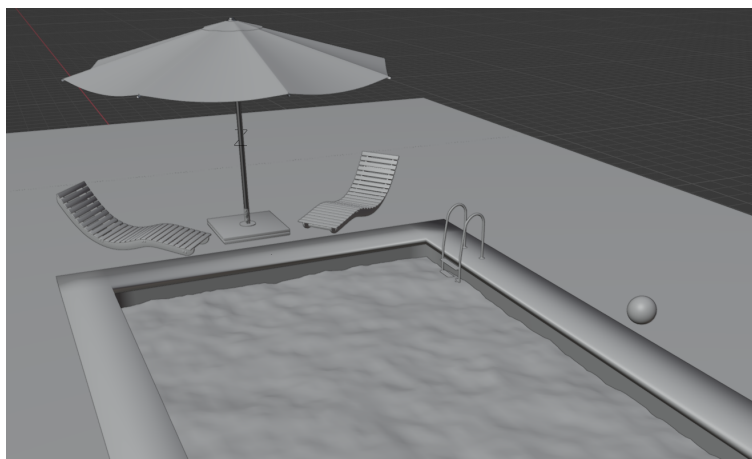
Pokračoval jsem vylepšením modelu bazénu. Nový model má tvar hrušky, vedou do něj schody ze stejného materiálu, a na stěnách má několik světel. Chvilí jsem experimentoval s jejich tvarem. Začal jsem dlouhými ovály, které byly částečně umístěny nad hladinou. Nakonec jsem světlo vymodeloval ze zploštělého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

S hruškovitým tvarem jsem nebyl spokojen, proto jsem se vrátil zpět na jednoduchý obdélníkový tvar (navíc u zaoblených tvarů se deformovala textura dlaždic). Okolo bazénu jsem umístil jednoduchou betonovou dlažbu. Původně byl v plánu trávník, ale ten jsem zavrhl z důvodu časové úspory při vykreslování scény bazénu. Do okolí jsem umístil několik volně dostupných objektů pod CC0 licencí citebk, abych scénu trochu oživil. Pro bazén jsem vytvořil dřevěný a mramorový rám a nakonec jsem 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 jed-

noduchého 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. Nevýhoda je v pozorovatelně delší době vykreslování - viz tabulky na konci práce 7.4. Nakonec jsem přidal i noc: scénu jsem osvětlil ještě slabším zdrojem světla než u zataženého počasí a do bazénu jsem přidal několik světél.

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.



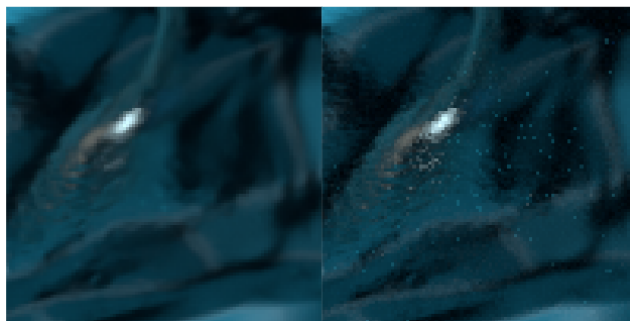
**Obrázek 4.1:** Scéna v Blenderu, bez textur

## Kapitola 5

### Rendering scény

#### 5.1 Srovnání Cycles Render a LuxCoreRender

Do Blenderu jsem nainstaloval experimentální verzi LuxCoreRenderu a upravil jsem materiál vody pro LuxCore. Následně jsem vyrenderoval dva snímky pomocí Cycles - u druhého jsem vypnul denoiser (256 vzorků, rozlišení 4K, stejné nastavení jako předchozí). Jediné místo s šumem, které jsem na obrázku našel, bylo kolem kovové tyče ve vodě u schodů 5.1. LuxCoreRender 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 LuxCoreRenderu je vykreslovat, dokud ho uživatel nezastaví, nastavil jsem maximální počet vzorků na 256. Vyrenderoval jsem dva obrázky, z nichž jeden měl vypnutý odšumovač, stejně jako u Cycles, viz obrázky. Jako další jsem nastavil počet vzorků na 512 - jak v Cycles, tak v LuxCore 5.2. Tentokrát jsem použil obousměrné sledování cest (místo jednosměrného), jediný problém je, že lze vykreslovat pouze pomocí CPU, čímž se několikanásobně prodloužila doba vykreslování 5.1.

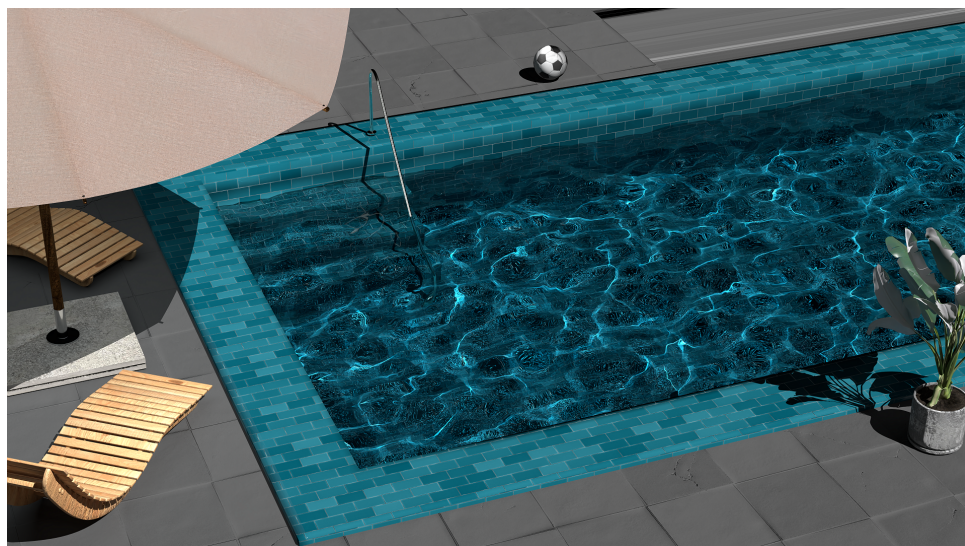


**Obrázek 5.1:** Odšumovač v Cycles - 256 vzorků.

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

**Tabulka 5.1:** Časy renderů Cycles a LuxCore. LuxCore na 512 vzorků byl renderován pouze pomocí CPU.

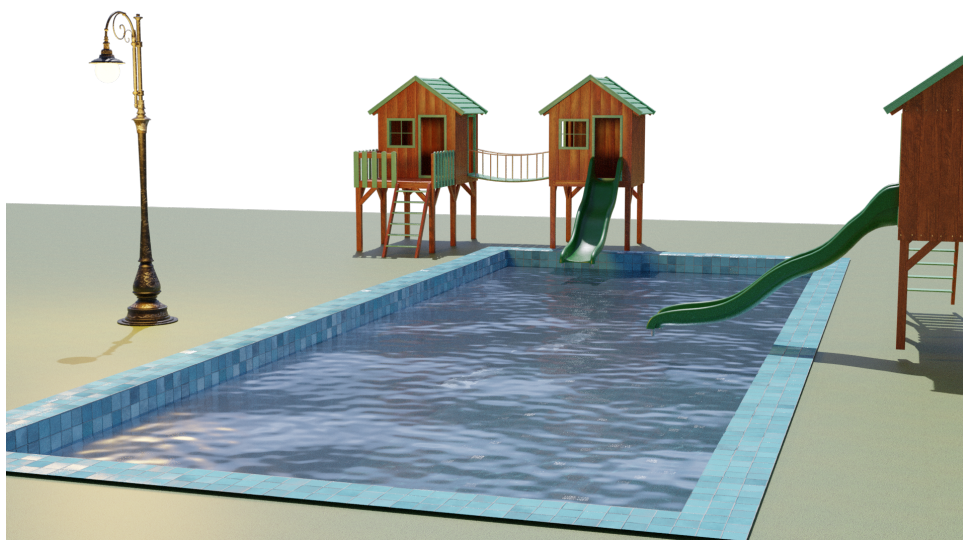
LuxCore vykresluje mnohem lepší kaustiky, což ale zabere mnohem více času. Nejlepších výsledků jsem dosáhl při za použití obousměrného sledování cest, ovšem za cenu zhruba čtyřnásobné doby vykreslování. 512 vzorků nebylo dostatek, aby se voda zbavila šumu. Samotná textura vody se mi v experimentální 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 snáz, dobře vykreslování je mnohonásobně nižší, což se nejvíc projeví na době renderování animací. Navíc jsem narazil na technické problémy při používání LuxCoreRenderu, kdy jsem byl nucen restartovat Blender a spustit vykreslování scény znovu.



**Obrázek 5.2:** LuxCore, 512 samples, bidirectional path tracing, denoised.

## 5.2 Vykreslování různých konfigurací

Jeden z prvních vykreslených snímků bazénu představuje obrázek 5.3, který byl renderován přes Cycles na 128 vzorků. Zkoušel jsem do scény umístit nějaké volně dostupné modely kolem bazénu, abych viděl odrazy ve vodě. Zde je voda tvořená pouze metodou bump mapping.

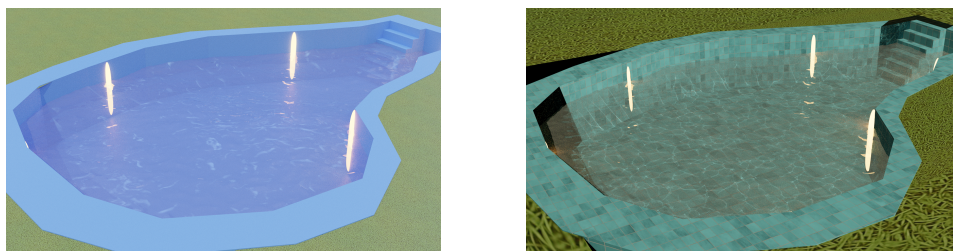


**Obrázek 5.3:** Jednoduchý bazén, Cycles

V dalších snímcích jsem změnil tvar bazénu a přidal světla dovnitř. Oba obrázky 5.4 byly vykresleny na 128 vzorků - jak v Cycles, tak v LuxCore. Všechny tyto rendery jsou ve FullHD (1920 × 1080) rozlišení a proběhly bez odšumovače (denoiser), takže v nich lze vidět zrnité oblasti.

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

Po znovu zprovoznění LuxCoreRendereru jsem vymodeloval klasický obdélníkový bazén s dřevěným lemem. Vykreslování jsem nechal běžet 42 minut, maximum světelných cest jsem nastavil na 32 a celkově proběhlo 7450 vzorků 5.5.



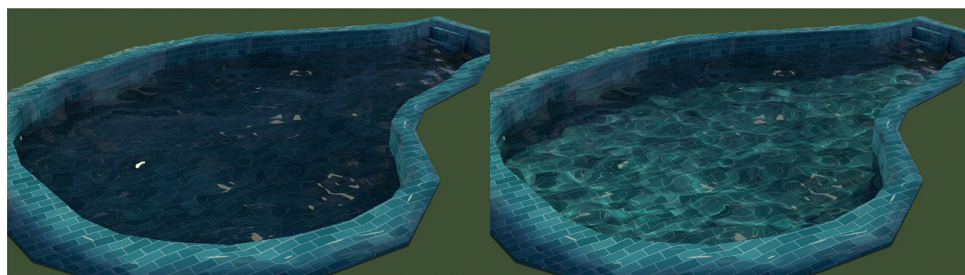
**Obrázek 5.4:** Cycles a LuxCore, složitější model.





**Obrázek 5.5:** LuxCore renderer - 42 min.

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



**Obrázek 5.6:** Kaustiky v Cycles.

Na konci práce, v poslední kapitole, je několik vykreslených obrázků finálního bazénu pro některé vybrané konfigurace. Všechny obrázky byly renderovány pomocí Cycles v rozlišení 4K. Počet vzorků jsem nastavil na 256. Snímky byly renderovány pomocí Cycles. Zvolil jsem 4K rozlišení, aby si uživatel mohl obrázek stáhnout a přiblížit, pokud by se chtěl podívat na detaily.

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.

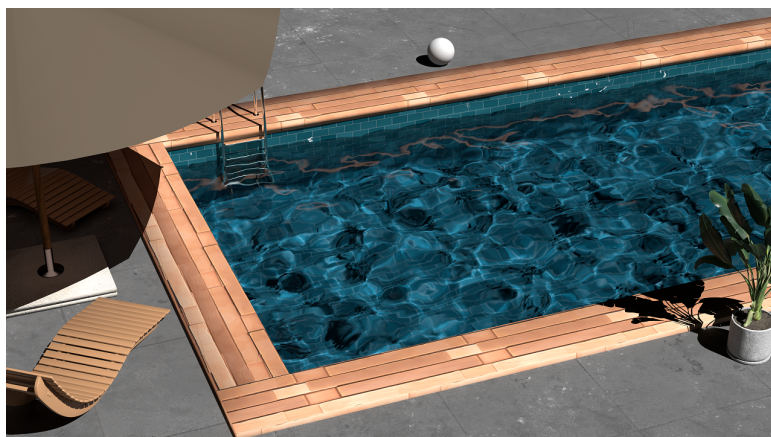
Časy jednotlivých renderů jsou uvedeny v tabulkách na konci práce. 7.2, 7.3, 7.4, 7.5 a 7.6.

Nakonec jsem vyrenderoval tři animaci bazénu, každá s jiným materiálem a v jiném počasí. Všechny animace jsou ve FullHD rozlišení a mají 60 snímků, ostatní nastavení je identické s předešlými obrázky. Doby vykreslení konfigurací jsou uvedeny v tabulce 7.1.

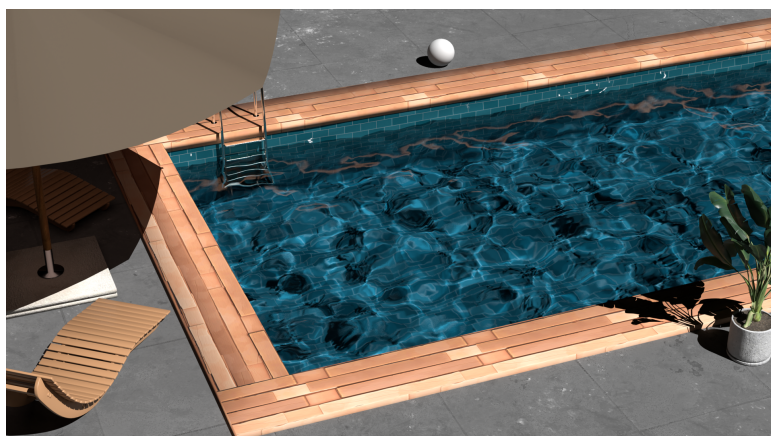
Jelikož jsem se v průběhu práce rozhodl, že se bazén bude vykreslovat lokálně, tak jsem se rozhodl změnit nastavení renderu, abych snížil celkovou dobu vykreslování. Pouze snížením rozlišení ze 4K na FullHD jsem se dostal na méně než šestinu času. Dále jsem snížil počet vzorků z 256 na 128, což ušetřilo zhruba třetinu času oproti předchozímu renderu. Nakonec jsem snížil počet Max Bounces v Light Paths na polovinu, se zanedbatelnými výsledky. Při zběžném porovnání obrázků jsou v nich pouze minimální rozdíly.

4K	198.15s
FullHD	29.23s
128 Vzorků	21.73s
Poloviční Max Bounces	20.35s

**Tabulka 5.2:** Porovnání času vykreslování



**Obrázek 5.7:** 4K rozlišení, 256 vzorků

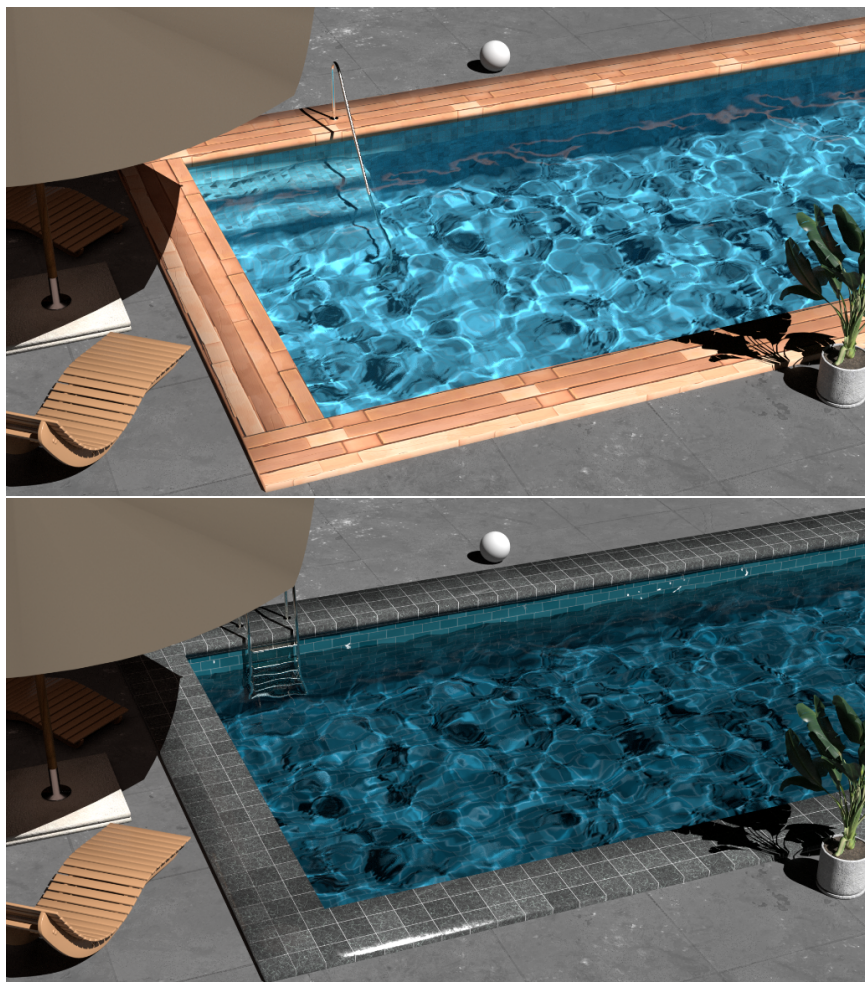


**Obrázek 5.8:** FullHD, 128 vzorků, polovina Max Bounces

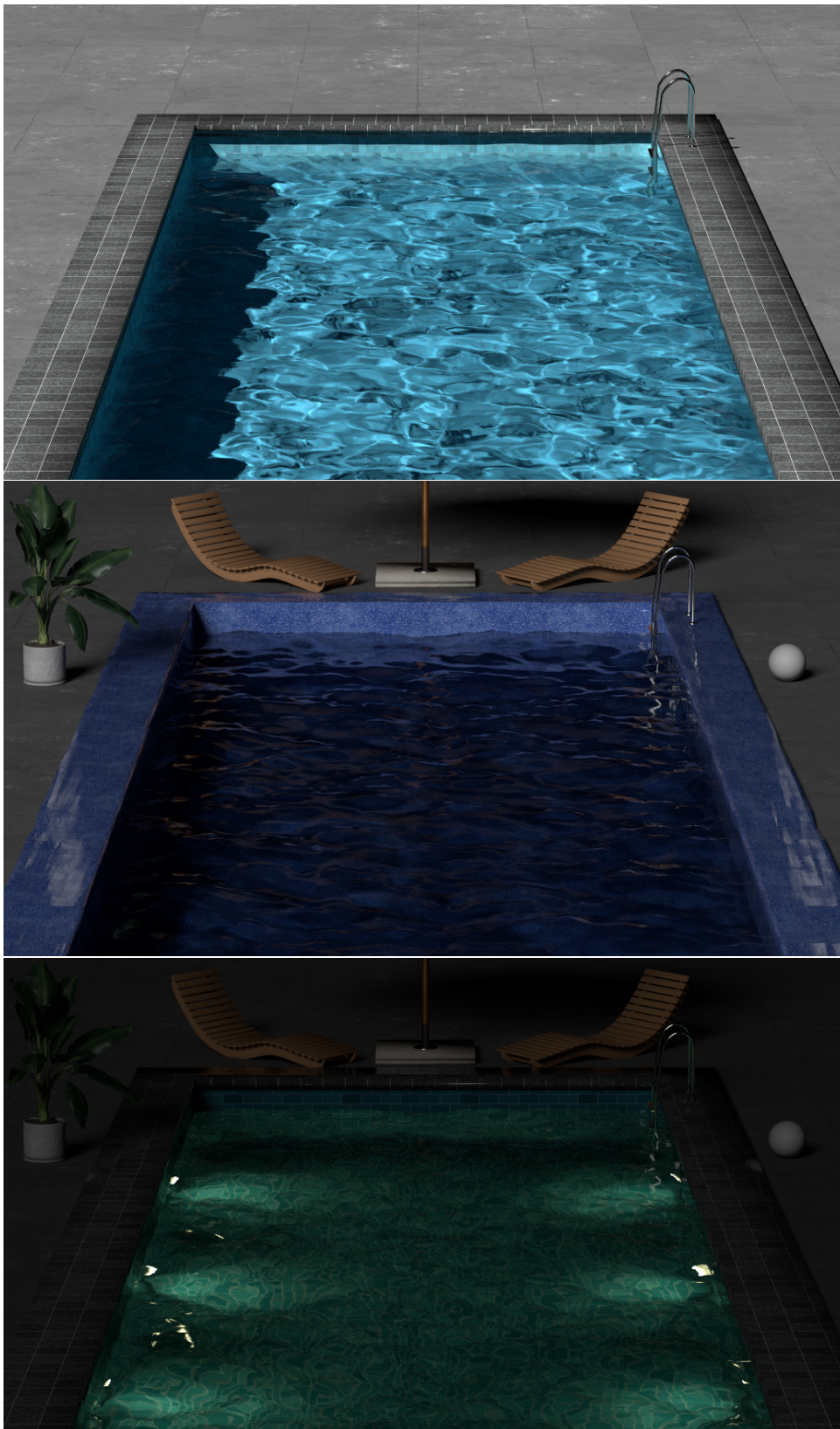
### 5.3 Jednotlivé konfigurace

Pro webovou stránku jsem vytvořil obrázky 54 různých konfigurací a 3 animace. Celkový čas vykreslování obrázků do aplikace byl 221 minut a 33 sekund. Animace se vykreslovaly celkem 169 minut a 10 sekund. Po spuštění vykreslování všech konfigurací ve webové aplikaci, Blender vygeneroval celkem 216 obrázků za 3 hodiny a 13 minut (FullHD rozlišení, 256 vzorků).

Ve finální aplikaci si uživatel může vybrat různé konfigurace a aplikace mu ukáže vykreslený 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, mramorový lem nebo žádný. Dekorace kolem bazénu lze vypnout a zapnout. Počasí lze vybrat buď slunečné, zatažené nebo noc. V noci je bazén osvětlen pouze světly umístěnými na stěnách bazénu. Nakonec uživatel vybírá pohled na bazén. Ten je možné zvolit přímo zepředu nebo z boku.



Obrázek 5.9:



Obrázek 5.10:



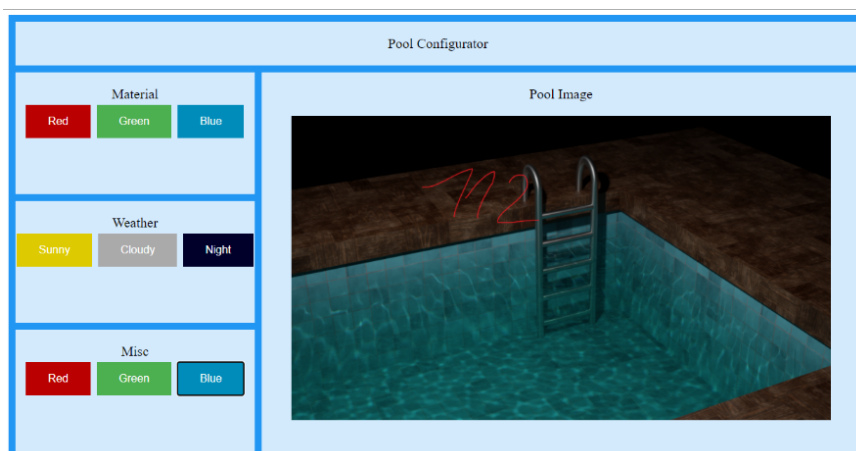
## Kapitola 6

### 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. Začal jsem jednoduchou webovou stránkou jazyce HTML s několika JavaScript funkcemi.

Při tvorbě uživatelského rozhraní jsem se inspiroval v konfigurátorech aut, jak jsem zmiňoval výše. Většinu obrazovky zabírá předvykreslený 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. Pomocí 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 předvykreslených obrázků. Tlačítka na uložení konfigurace jsem přemístil pod obrázek bazénu abych efektivněji využil prostor.

Další krok byl generování konfigurace bazénu za běhu. Ideální řešení je, aby webová aplikace poslala požadavek na vzdálený server, kde se spustí vykreslování v Blenderu. Po vykreslení dané konfigurace server nahraje obrázek a uživatel si ho bude moct kdykoliv zobrazit. Postupně by se tedy vygenerovali všechny konfigurace a zůstaly by v paměti. Bohužel toto řešení je momentálně nad mé schopnosti, proto jsem se zaměřil na vykreslování bazénu na lokálním počítači. Webovou stránku jsem převedl na HTML aplikaci (HTA), která může spustit vykreslování v Blenderu přes příkazovou řádku. Pod obrázkem bazénu je tlačítko, které spustí vykreslování všech konfigurací v Blenderu, ale jen těch, které ještě nebyly vygenerované.

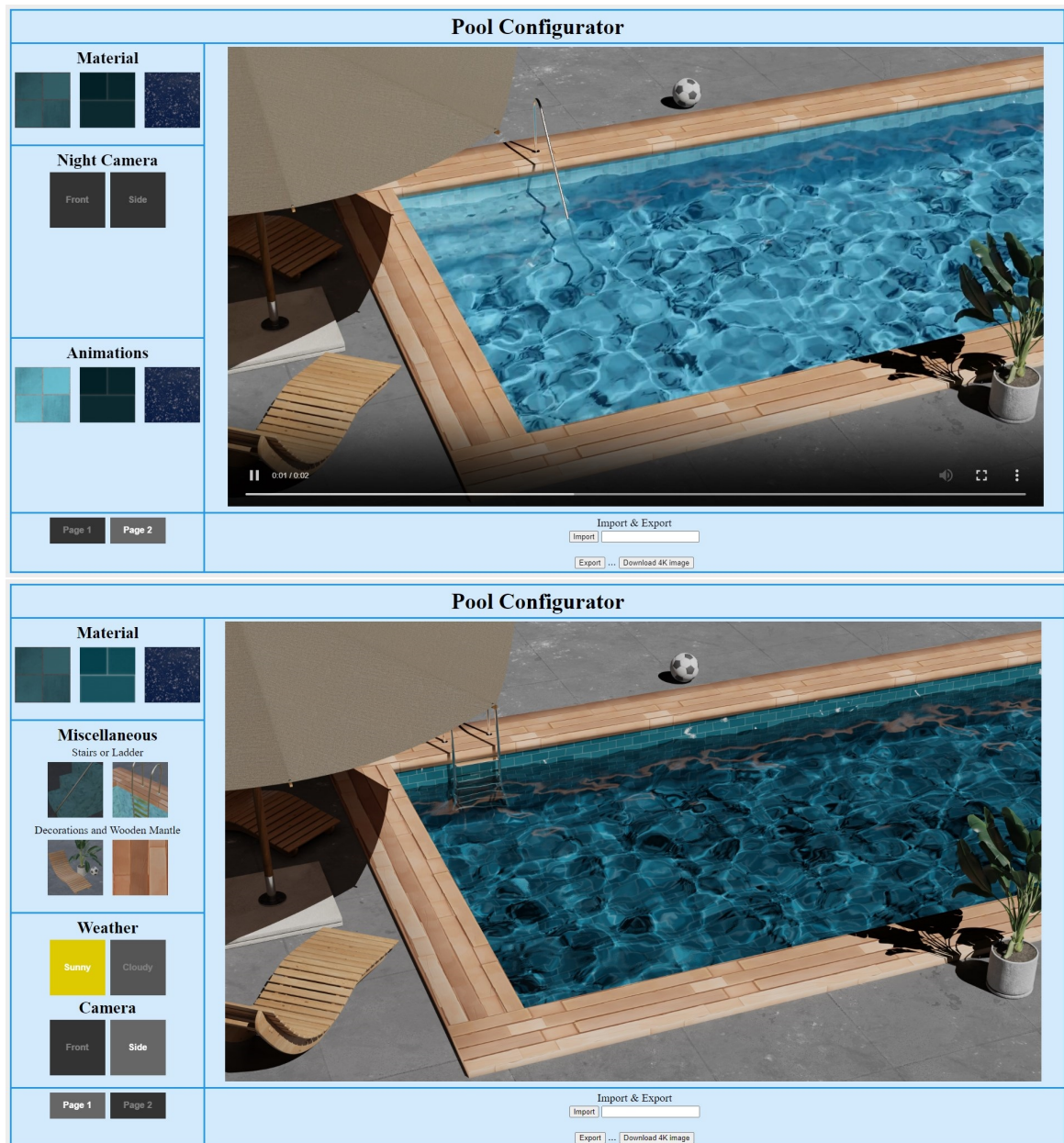
## 6.2 Finální vzhled aplikace

U webové stránky jsem zůstal 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.

HTML aplikace ztratila část funkcionalit kaskádových stylů, tlačítka jsou přesunuta do dvou řádků nad obrázkem bazénu a nemění barvu po výběru. V aplikaci si lze vybrat konfiguraci bazénu pro:

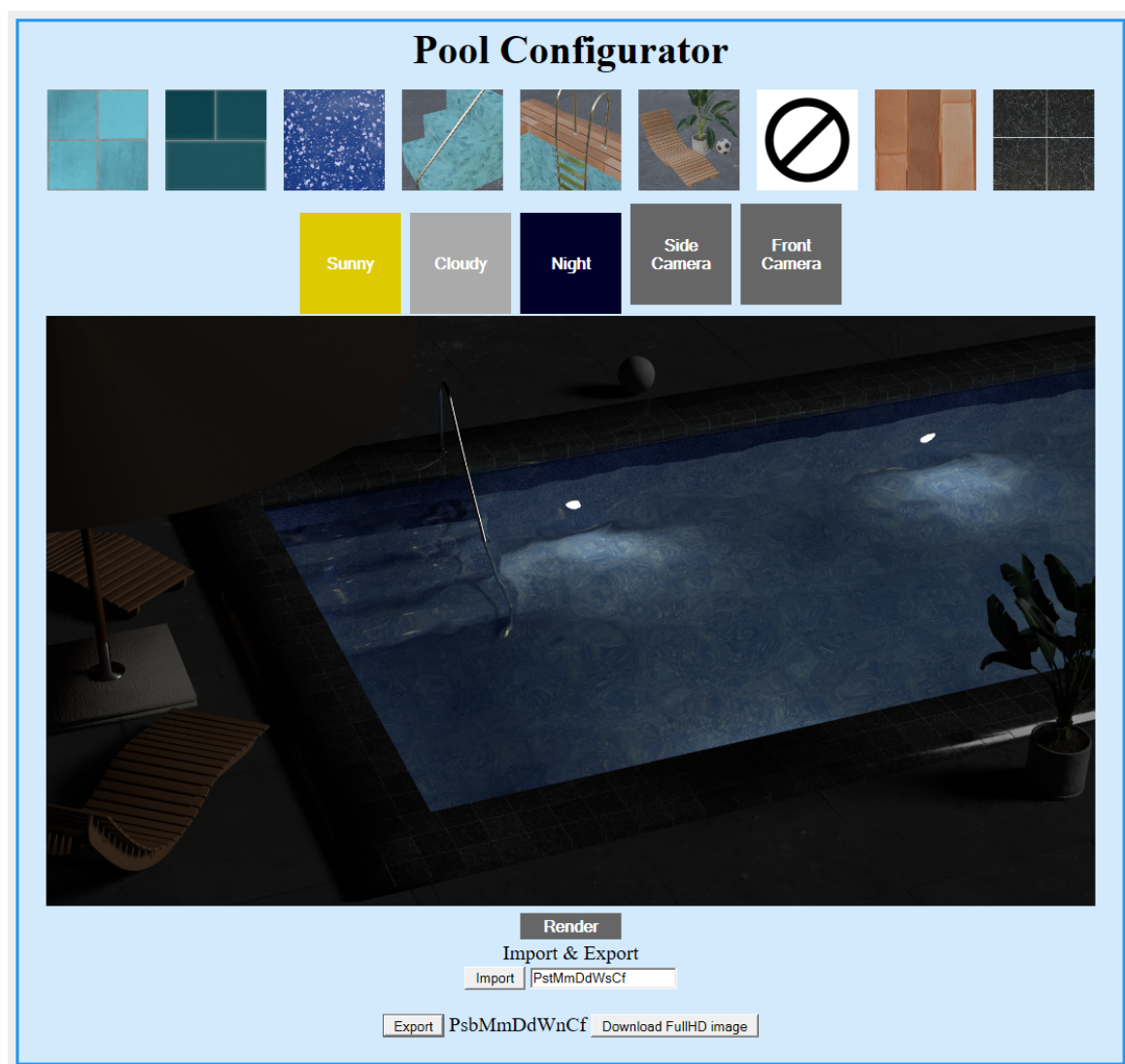
- 3 různé materiály,
- 2 vstupy do bazénu,
- 3 typy lemů bazénu,
- zda-li se mají zobrazovat dekorace okolo bazénu,
- 3 denní doby
- 2 pohledy na bazén.

To dohromady dává 18 různých konfigurací samotného bazénu a celkem 216 možných obrázků.



Obrázek 6.2: Finální vzhled konfiguratoru.





Obrázek 6.3: Finální vzhled HTA konfigurátoru.

## 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 s Cycles rendererem. Popsal jsem postup vymodelování scény s bazénem a na jaké problémy jsem narazil. Poté jsem představil výsledky vykreslených scén, jak z LuxCoreRendereru, tak z Cycles. Postup tvorby aplikace jsem probral v poslední kapitole. Na konci práce se nachází tabulka s časy vykreslování obrázků bazénu a některé tyto obrázky.

Konfigurátor by šel rozhodně vylepšit, jak samotná aplikace, tak scéna v Blenderu. Aplikaci by jistě prospěl modernější design a více možností konfigurace. 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.

Vykreslování scény by ideálně mělo probíhat na vzdáleném serveru. Webová aplikace by poslala požadavek na server pro danou konfiguraci. Server by se podíval, jestli už má bazén pro tuto konfiguraci vygenerován - pokud ano, poslal by zpět daný obrázek. V opačném případě by spustil vykreslování a obrázek by poslal až by byl vygenerován. Mohl by posílat i průběžně, aby uživatel viděl, že se obrázek teprve vytváří.

Momentálně uživatel musí mít nainstalován Blender a také musí mít dostatečný hardware pro vykreslování dané scény. Vykreslování lze rozhodně zrychlit, ovšem za cenu kvality obrázku. Z testování poměrně dobře funguje pouze 64 vzorků za slunečného dne. Nejrychlejší je ovšem snížení rozlišení pouze na HD (zde jeden obrázek zabere pouze lehce přes 10 sekund).

## 7.1 Tabulky a výsledné vykreslené obrázky

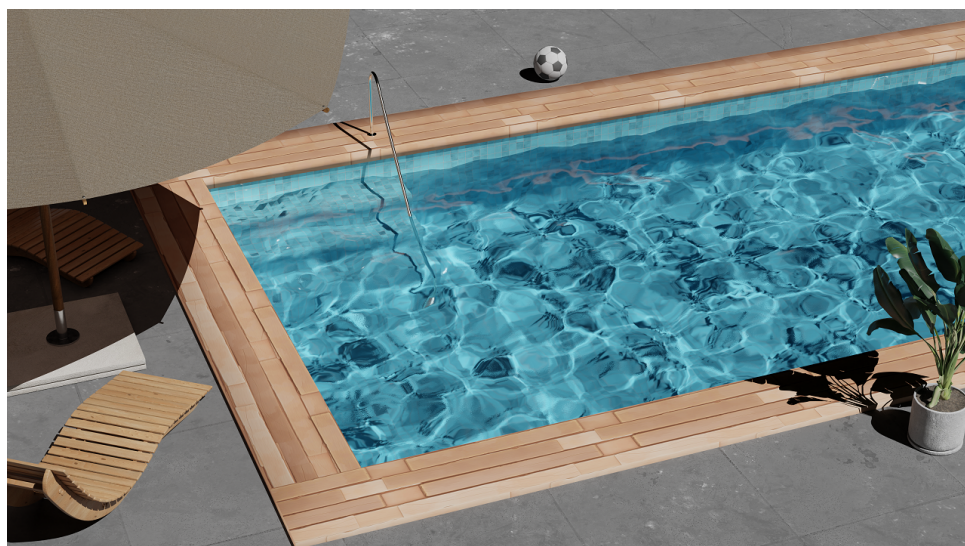
Časové údaje v tabulkách jsou ve formátu minuty:sekundy. Použitý hardware: AMD Ryzen 5 5600X, NVIDIA GeForce RTX 3070. Nastavení renderu: rozlišení 4K, 256 vzorků, Light Paths - Max Bounces: Total=16, Diffuse=4, Transmission=12, Volume=4, Transparent=8.

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 7.1:** Časy vykreslování 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 7.2:** Časy vykreslování pro různé konfigurace při slunečném počasí.



**Obrázek 7.1:** 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 7.3:** Časy vykreslování 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 7.4:** Časy vykreslování pro různé konfigurace za horší viditelnosti.



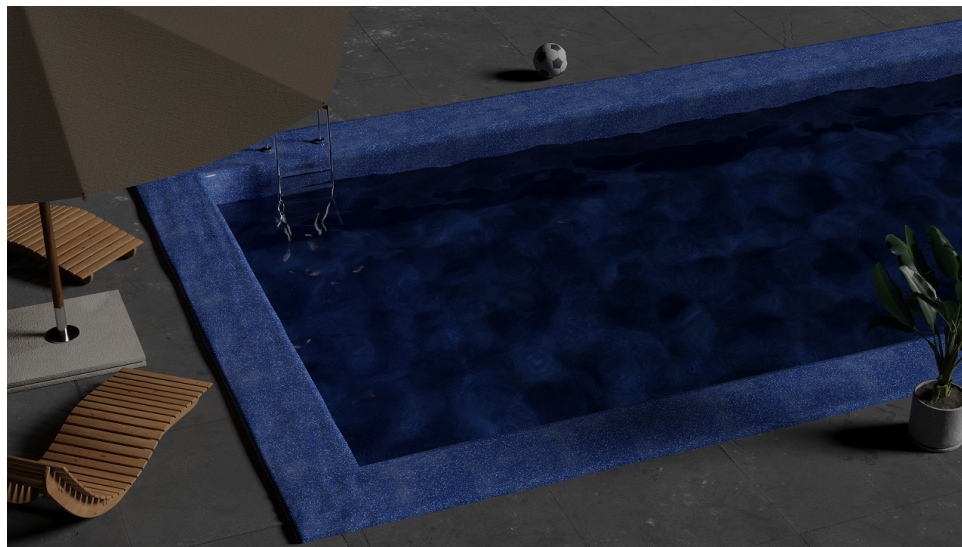
**Obrázek 7.2:** Modrozelené kachličky, slunečný den

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

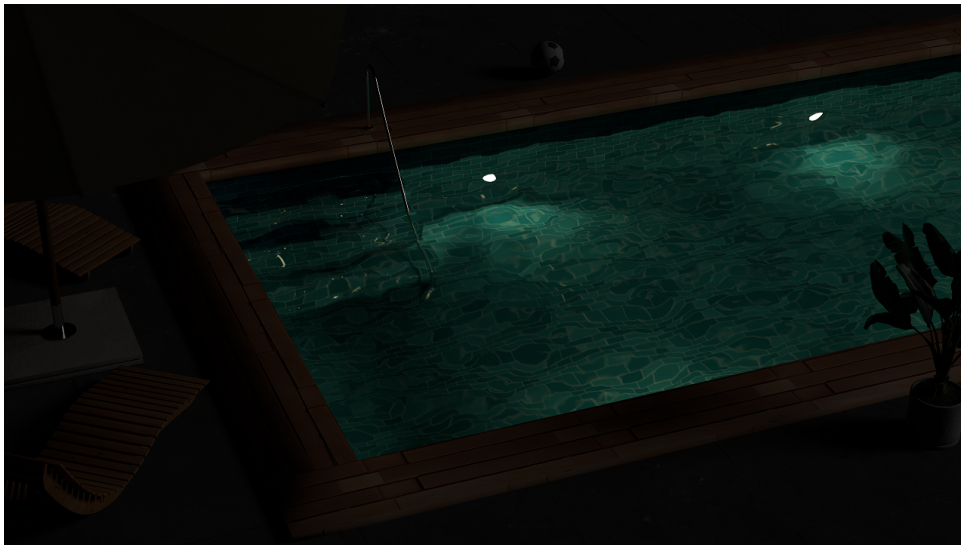
**Tabulka 7.5:** Časy vykreslování pro různé konfigurace za horší viditelnosti.

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

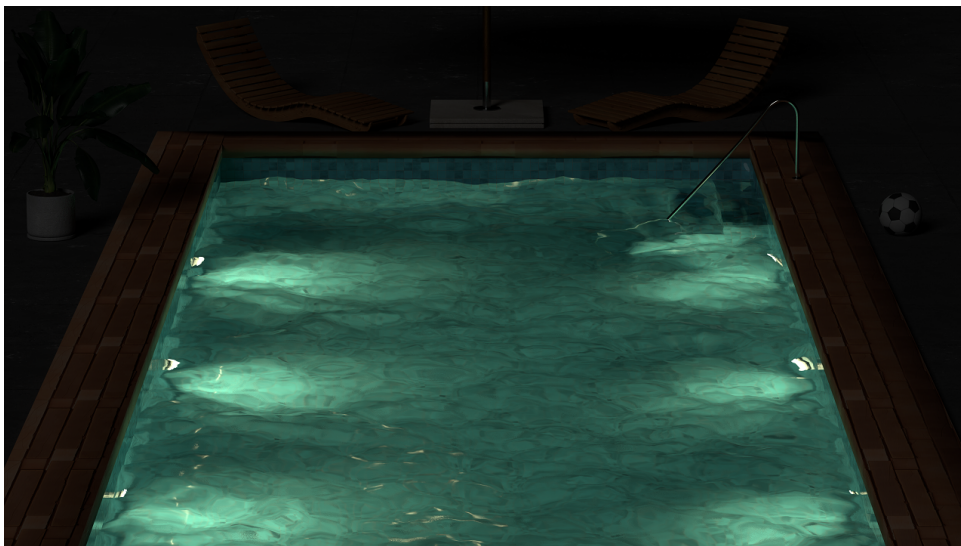
**Tabulka 7.6:** Časy vykreslování pro různé konfigurace v noci.



**Obrázek 7.3:** Modrobílý vzor, nižší viditelnost



**Obrázek 7.4:** Modrozelené kachličky, noční osvětlení



**Obrázek 7.5:** Tyrkysová kachličky, noční osvětlení





## Literatura

- [1] Matt Pharr, Wenzel Jakob, and Greg Humphreys, *Physically Based Rendering: From Theory To Implementation*, Third Edition, 2018, online, [www.pbr-book.org](http://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](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/hdri.s>.
- [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/](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>.
- [12] *The Houdini Ocean Toolkit*, 2023, <https://code.google.com/archive/p/houdini-ocean-toolkit/>, <https://github.com/eloop/hot>.
- [13] *"Practical Path Guiding" in Production*, 2023, [https://cgg.mff.cuni.cz/~jirka/path-guiding-in-production/2019/presentations/Guiding\\_In\\_Production\\_Course\\_s2019-2019-08-07-ppg-in-production.pdf](https://cgg.mff.cuni.cz/~jirka/path-guiding-in-production/2019/presentations/Guiding_In_Production_Course_s2019-2019-08-07-ppg-in-production.pdf)