

VYSOKÁ ŠKOLA POLYTECHNICKÁ JIHLAVA

Aplikovaná Informatika

**SROVNÁNÍ VYBRANÝCH SOFTWAREVÝCH
NÁSTROJŮ PRO 3D MODELOVÁNÍ**

Bakalářská práce

Autor práce: David Linha

Vedoucí práce: doc. Ing. Zbyněk Bureš, Ph.D.

Jihlava 2026

Vysoká škola polytechnická Jihlava

Tolstého 16, 586 01 Jihlava

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Autor práce:	David Linha
Studijní program:	Aplikovaná informatika
Obor:	Aplikovaná informatika
Garant studijního programu:	doc. Ing. Lenka Kuklišová Pavelková, Ph.D.
Název práce:	Srovnání vybraných softwarových nástrojů pro 3D modelování
Vedoucí práce:	doc. Ing. Zbyněk Bureš, Ph.D.
Cíl práce:	Cílem práce je porovnat dva vybrané programy pro 3D modelování, Blender a 3DSMax, se zaměřením na jejich schopnosti vytvářet detailní modely a simulace. V obou prostředích bude podle skutečné předlohy vytvořen detailní 3D model bitevní lodě, včetně video a zvukových efektů, a simulace okolní vody. Dále bude zkoumána schopnost obou programů simulovat střelbu a destrukci lodě ve vodním prostředí. Výstupem práce bude rovněž podrobná analýza předností a slabin obou programů a praktické rady pro řešení podobných úloh.

Abstrakt

Bakalářská práce se zabývá srovnáním dvou nástrojů pro 3D modelování a simulace v Blenderu a Autodesk 3DS Max. Cílem práce je analyzovat jejich schopnosti vytvářet detailní 3D modely a provádět fyzikální simulace, přičemž zvláštní pozornost je věnována modelování bitevní lodě a simulacím vody, střelby a destrukce ve vodním prostředí.

Klíčová slova

3D modelování, Blender, 3DS Max, fyzikální simulace, vizualizace, počítačová grafika.

Abstract

Bachelor's thesis focuses on comparing two tools for 3D modeling and simulation in Blender and Autodesk 3DS Max. The aim of the thesis is to analyze their capabilities in creating detailed 3D models and conducting physical simulations, with special emphasis on modeling a battleship and simulating water, shooting, and destruction in a water environment

Keywords

3D modeling, Blender, 3DS Max, physical simulations, visualization, computer graphics.

Prohlašuji, že předložená bakalářská práce je původní a zpracoval/a jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem v práci neporušil/a autorská práva (ve smyslu zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů, v platném znění, dále též „AZ“).

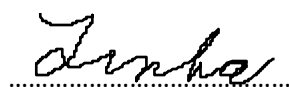
Byl/a jsem seznámen/a s tím, že na mou bakalářskou práci se plně vztahuje **AZ**, zejména § 60 (školní dílo).

Podle § 47b zákona o vysokých školách souhlasím se zveřejněním své práce podle Směrnice pro vedení, vypracování a zveřejňování závěrečných prací na VŠPJ, a to bez ohledu na výsledek obhajoby.

Beru na vědomí, že VŠPJ má právo na uzavření licenční smlouvy o užití mé bakalářské práce a prohlašuji, že **souhlasím** s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom/a toho, že užití své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem VŠPJ, která má právo ode mě požadovat přiměřený příspěvek na úhradu nákladů, vynaložených vysokou školou na vytvoření díla (až do jejich skutečné výše), z výdělku dosaženého v souvislosti s užitím díla či poskytnutím licence.

V Jihlavě dne 15.4.2026



Podpis studenta/ky

Poděkování

Rád bych poděkoval doc. Ing. Zbyňku Burešovi, Ph.D., za jeho vedení mé bakalářské práce.

Dále bych chtěl poděkovat Bc. Marku Růžičkovi a Bc. Ondřeji Liškovi za cenné rady poskytnuté při tvorbě mé bakalářské práce.

Obsah

Seznam obrázků.....	7
Úvod.....	8
1 Teoretická část.....	9
1.1 Úvod do 3D modelování.....	9
1.2 Simulace ve 3D prostředí.....	10
1.3 Blender.....	12
1.4 Autodesk 3DS MAX.....	19
2 Praktická část.....	24
2.1 Výběr předlohy a metodika modelování.....	24
2.2 Tvorba modelu v Blenderu.....	24
2.3 Tvorba modelu v 3DS MAX.....	26
2.4 Definice povrchových vlastností a PBR materiálů.....	28
2.5 Simulace vodního prostředí a interakce plavidla v Blenderu.....	28
2.7 Destrukce v Blenderu.....	30
2.8 Destrukce v 3DS MAX.....	34
2.9 Simulace vodního prostředí a limity modulu Fluid ve 3DS MAX.....	34
2.10 Volumetrické simulace a efekty (VFX).....	36
2.11 Finální příprava scény v Blenderu.....	37
2.12 Finální příprava scény v 3DS MAX.....	37
2.13 Optimalizace výpočetního času a nastavení Raytracingu.....	38
2.14 Finální rendering a kompletace audiovizuálních výstupů.....	38
Závěr.....	40
Seznam použité literatury.....	42

Seznam obrázků

Obrázek 1: Layout.....	14
Obrázek 2: View preference Zdroj: vlastní práce.....	15
Obrázek 3: Outliner.....	15
Obrázek 4: Properties editor.....	16
Obrázek 5: Timeline Zdroj: vlastní práce.....	17
Obrázek 6: Tool shelf Zdroj: vlastní práce.....	17
Obrázek 7: Přispůsobení halvního prostoru.....	18
Obrázek 8: Trasnform panel.....	19
Obrázek 9: Viewporty.....	21
Obrázek 10: Command panel.....	21
Obrázek 11: Modifier stack.....	22
Obrázek 12: Main toolbar.....	22
Obrázek 13: Material editor.....	23
Obrázek 14: Dělo_Blender.....	25
Obrázek 15: Zád' lodi_ Blender.....	26
Obrázek 16: Trup lodi v 3DS MAX.....	27
Obrázek 17: "Falešný" oceán.....	30
Obrázek 18: Dynamic paint_canvas.....	31
Obrázek 19: Dynamic paint_brush.....	31
Obrázek 20: Test cell fracture.....	32
Obrázek 21: Cell fracture.....	33
Obrázek 22: Simulace vody.....	35
Obrázek 23: Příprava na simulaci vody.....	36
Obrázek 24: Simulace výbuchu.....	37

Úvod

3D modelování a simulace se v posledních letech čím dál více používá ve všech možných odvětvích průmyslu, nejvíce se však používá ve filmovém a herním průmyslu. Mezi nejrozšířenější volně dostupné programy pro 3D modelování a simulace patří Blender, Autodesk 3DS Max je profesionální program pro tvorbu 3D grafiky, přičemž oba nabízejí širokou škálu funkcí.

Cílem mé bakalářské práce je kompletní porovnání těchto dvou nástrojů.

Práce bude rozdělena do dvě fáze. Nejprve budou oba programy teoreticky popsány a bude zhruba popsána historie a nástroje obou programů. Dále bude následovat praktická část, kde bude podle skutečné předlohy vytvořen detailní model bitevní lodě v obou prostředích. V praktické části se zaměřím na efektivitu modelování, možnosti texturování a osvětlení. Po dokončení modelu bude následovat simulace vody, střelby a destrukce ve vodním prostředí, která umožní praktické porovnání schopností obou programů v oblasti fyzikálních simulací.

Výstupem mé práce bude podrobná analýza předností a slabín obou nástrojů, včetně praktických rad pro řešení podobných problémů. Cílem je také poskytnout uživatelům praktický návod, jak efektivně využít oba programy pro tvorbu realistických 3D scén s pokročilými simulacemi.

1 Teoretická část

V teoretické části budou popsány oba programy včetně jejich primárních funkcí a nástrojů.

1.1 Úvod do 3D modelování

3D modelování je proces tvorby trojrozměrných objektů na počítači pomocí specializovaných softwarových nástrojů. Výsledné 3D modely mohou být využity v široké škále aplikací, od filmového a herního průmyslu až po vědu, architekturu a průmyslový design (Chopine, 2011). Taková technika umožňuje tvůrcům přenést jejich kreativní vize do virtuálního prostoru.

1.1.1 Co je 3D modelování?

3D modelování zahrnuje vytváření virtuálních objektů, které mají trojrozměrné vlastnosti, ku, šířku a hloubku. Objekty jsou tvořeny základními prvky, jako jsou vrcholy (Vertices) Základní body definující geometrii modelu.

Hrany (Edges): Spojnice mezi vrcholy, které určují tvar modelu.

Plochy (Faces): Trojúhelníky nebo čtyřúhelníky, které tvoří povrch modelu.

3D modelování vyžaduje nejen technické znalosti, ale také estetické cítění a pochopení prostorových vztahů. Tvůrce musí zohlednit detailnost modelu, jeho účel a možnosti softwaru, ve kterém pracuje.

1.1.2 Proces 3D modelování

Samotný proces 3D modelování obsahuje několik základních kroků.

1. Koncept a návrh: Proces začíná vytvořením konceptu, například skici nebo náčrtu, který určuje, jak bude model vypadat.
2. Tvorba základní geometrie: Tvůrce začne s jednoduchými tvary, jako jsou kostky, koule nebo válce, které jsou postupně upravovány.
3. Detailní úpravy: Přidávají se detaily, například hrany, otvory nebo reliéfy, aby model vypadal realisticky.
4. Texturování a materiály: Na model se aplikují textury a materiály, které určují jeho vzhled – barvu, odrazivost, průhlednost atd.
5. Osvětlení: Pro finální vizualizaci se nastaví světla, která simulují přirozené nebo umělé osvětlení.
6. Renderování: Hotový model se převede do obrázku nebo animace pomocí renderovacího enginu.

1.1.3 Význam 3D modelování

3D modelování se stalo nepostradatelným nástrojem v mnoha oborech:

Filmový a herní průmysl: Tvorba postav, prostředí a vizuálních efektů. Například seriál jako *Love, Death & Robots* nebo hry jako *The Sims* využívají 3D modely pro realistické zobrazení.

Průmyslový design: Vytváření prototypů produktů před jejich fyzickou výrobou.

Architektura: Vizualizace stavebních projektů a interiérového designu.

Věda a medicína: Modelování molekul, orgánů nebo simulací fyzikálních jevů.

3D modelování umožňuje tvůrcům experimentovat s návrhy bez nutnosti fyzických materiálů, což šetří čas a náklady.

1.1.4 Typy 3D modelování

Existuje několik přístupů k 3D modelování, které jsou používány v závislosti na účelu modelu:

1. **Polygonální modelování:** Nejrozšířenější metoda, kde jsou objekty tvořeny z polygonů (většinou trojúhelníků nebo čtyřúhelníků). Což je vhodné pro hry a animace.
2. **NURBS modelování:** Využívá matematické křivky pro přesné a hladké povrchy. Často se používá v průmyslovém designu a automobilovém průmyslu.
3. **Sochařství (Sculpting):** Technika připomínající práci sochaře, kde se model vytváří plynulým přidáváním a odebráním materiálu. Vhodné pro organické tvary, například zvířata, lidí apod. Nejčastěji se v tomto typu modelování používá grafický tablet se stylusem.
4. **Parametrické modelování:** Používá se v primárně CAD (Computer-Aided Design) softwarech, kde je možné model snadno měnit pomocí parametrů.

1.1.5 Budoucnost 3D modelování

S rostoucím výkonem počítačů a rozvojem technologií, jako je virtuální realita nebo strojové učení, se očekává, že 3D modelování bude hrát stále důležitější roli. Automatizace některých částí procesu, například generování textur nebo animací pomocí umělé inteligence, může práci zrychlit a zpřístupnit širší veřejnosti.

Tato kapitola poskytuje teoretický základ pro pochopení toho, co je 3D modelování, a připravuje půdu pro hlubší srovnání nástrojů Blender a 3DS Max, které následuje v dalších

1.2 Simulace ve 3D prostředí

Simulace je klíčovou součástí moderní 3D tvorby, která umožňuje realistické zobrazení fyzikálních jevů ve virtuálním prostředí. Simulace zahrnují například pohyb tekutin, chování pevných těles při nárazu, gravitaci, destrukci objektů a interakce mezi různými materiály. Díky simulacím je možné vytvářet scény, které působí autenticky a odpovídají reálnému světu, aniž by bylo nutné tyto jevy fyzicky reprodukovat.

1.2.1 Co je simulace?

Ve 3D modelování a animaci označuje simulace proces výpočtu a vizualizace fyzikálních vlastností a interakcí ve virtuálním prostředí. Na základě matematických modelů a algoritmů jsou v simulacích napodobovány přírodní jevy, jako je pohyb vody, deformace materiálů, gravitační síly nebo chování částic.

Například simulace vody v 3D programech zahrnuje sledování toho, jak kapalina reaguje na pohyb objektů, sílu gravitace nebo vnější síly, jako je vítr. Podobně simulace destrukce sleduje, jak se materiál rozpadá při nárazu nebo výbuchu.

1.2.2 Druhy simulací ve 3D prostředí

Je několik druhů simulací ve 3D prostředí:

1. Simulace tekutin

Slouží k zobrazení chování kapalných látek, jako je voda, olej nebo krev. Používá se například pro vytvoření realistických scén, jako jsou vlny, vodopády nebo interakce vody s objekty. Nástroje jako Mantaflow v Blenderu nebo Phoenix FD v 3DS Max jsou speciálně navrženy pro simulaci tekutin.

2. Simulace částic

Umožňuje vytvářet efekty, jako jsou kouř, prach, déšť nebo exploze. Částicové systémy sledují pohyb tisíců malých bodů, které reagují na vnější síly, například vítr nebo gravitaci.

3. Simulace destrukce

Modeluje, jak se objekty lámou, tříští nebo rozpadají pod vlivem nárazů nebo výbuchů. Často se využívá při tvorbě scén destrukce budov, automobilů nebo bitevních lodí.

4. Simulace látek a měkkých těles

Umožňuje realistické chování materiálů, jako jsou látky, polštáře nebo gumové objekty. Takové simulace zahrnují deformace, skládání a interakce s jinými objekty.

1.2.3 Význam simulací ve 3D

Simulace přináší do 3D tvorby nový rozměr realismu. Umožňují:

Vytváření věrohodných vizuálních efektů: Například realistické vlny na moři, kouř po explozi nebo realistický pohyb oblečení postav.

Úsporu času a nákladů: Simulace ve virtuálním prostředí nahrazuje fyzické experimenty, které by jinak vyžadovaly drahé vybavení a materiály.

Experimentování: Tvůrci mohou testovat různé scénáře a jejich vliv na scénu, například různé rychlosti větru nebo síly nárazu.

1.2.4 Simulace v Blenderu a 3DS Max

Oba programy nabízejí robustní nástroje pro simulaci fyzikálních jevů:

Blender

Blender obsahuje systém Mantaflow, který umožňuje pokročilé simulace tekutin, kouře a ohně. Nástroj Rigid Body Physics se využívá pro simulaci pevných těles a destrukcí. Blender také nabízí jednoduché částicové systémy, které lze využít k tvorbě efektů, jako je déšť nebo jiskry.

3DS Max

3DS Max využívá různé pluginy, jako je Phoenix FD a Particle Flow, které poskytují pokročilé možnosti simulací tekutin a částic. Integrované nástroje pro simulaci pevné fyziky umožňují precizní práci s destrukčními efekty. 3DS Max je známý svou schopností zvládat složité simulace v profesionálních produkcích.

1.3 Blender

Blender je svobodný a otevřený software pro počítačovou grafiku, který se používá k vytváření animovaných filmů, vizuálních efektů, uměleckých děl, třírozměrných tištěných modelů, pohyblivé grafiky, interaktivních třírozměrných aplikací, virtuální reality a videoher.

1.3.1 Historie

Historie Blenderu sahá až do počátku 90. let, kdy byl vytvořen jako interní nástroj pro nizozemské animační studio NeoGeo. Za jeho vznikem stojí nizozemský výtvarník a programátor Ton Roosendaal.

První verze Blenderu byla vytvořena v roce 1994 a byla zaměřena na uspokojení potřeb animačního studia. Roosendaal vkládal do Blenderu své zkušenosti a touhu vytvořit nástroj, který by umožnil efektivní a kreativní práci s 3D grafikou. Blender se rychle rozvíjel a získal si pozornost díky svým inovativním funkcím a přístupu. Přelomovým okamžikem v historii Blenderu bylo rozhodnutí učinit z něj open-source projekt. V roce 2002 byla spuštěna crowdfundingová kampaň, která umožnila získat potřebné finanční prostředky a Blender byl vydán pod licenci GNU General Public License. Tím se otevřely dveře pro širokou komunitu nadšenců, kteří se mohli podílet na vývoji a vylepšování tohoto softwaru.

Jedním z klíčových faktorů úspěchu Blenderu je jeho silná komunita uživatelů a vývojářů z celého světa. Komunitou řízené projekty, jako jsou Open Movies, umožňují umělcům spolupracovat a vytvářet ambiciózní díla. Blender Foundation, nezisková organizace založená Tonem Roosendaalem, poskytuje podporu komunitě a koordinuje vývoj Blenderu.

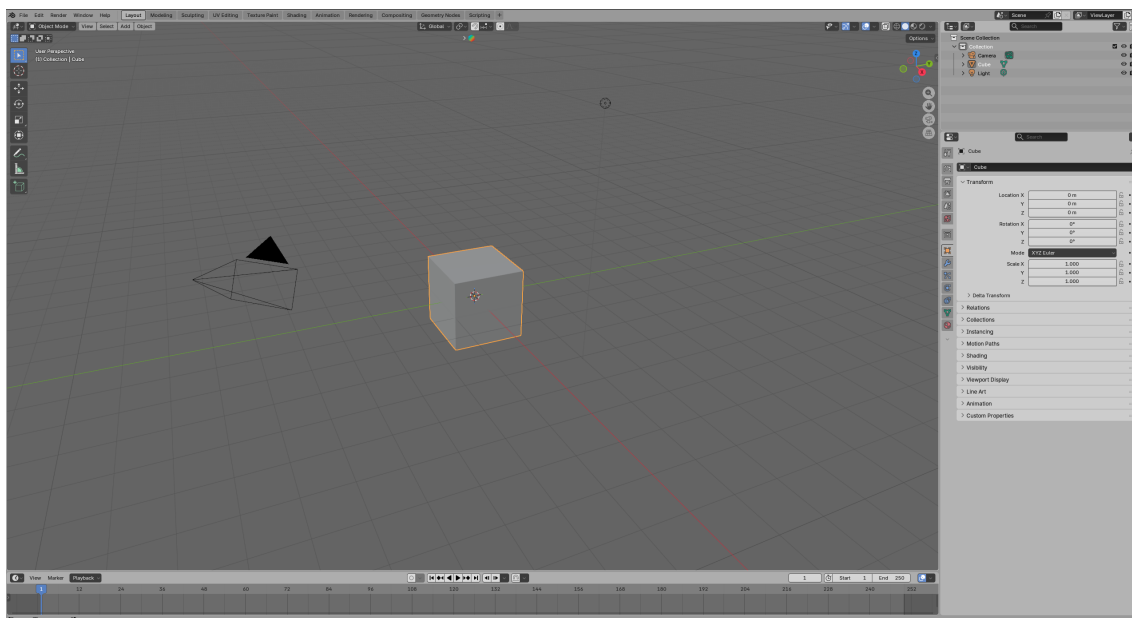
Dnes je Blender využíván profesionály i amatéry, ale převážně amatéry hlavně kvůli tomu že Blender je zdarma, používá se v různých oborech, jako je film, animace, architektura, design a vizuální efekty. Jeho open-source charakter umožňuje neustálý vývoj a přizpůsobování se novým technologiím a potřebám uživatelů. Budoucnost Blenderu vypadá slibně. S rostoucí popularitou 3D grafiky a virtuální reality se očekává, že Blender bude hrát stále významnější roli. Jeho otevřený charakter a neustálý vývoj z něj činí ideální nástroj pro všechny, kteří chtějí tvořit úžasné věci ve 3D světě. (Blender, 2024)

1.3.2 Nástroje a funkce

Blender disponuje velkou škálou nástrojů a funkcí od modelování až po editaci videa či zvuku. Blender řadí své nástroje do jednotlivých záložek, ale spoustu funkcí a nástrojů lze používat v úvodní záložce Layout.

1.3.3 Layout

Záložka **Layout** je základním pracovním prostředím Blenderu, které umožňuje uživatelům spravovat scénu, manipulovat s objekty. Záložka Layout je výchozím bodem pro většinu činností a nabízí přehledné a přizpůsobitelné rozhraní s klíčovými nástroji pro práci ve 3D prostoru. Záložka Layout je rozdělena na několik hlavních částí, které společně tvoří intuitivní pracovní prostředí:

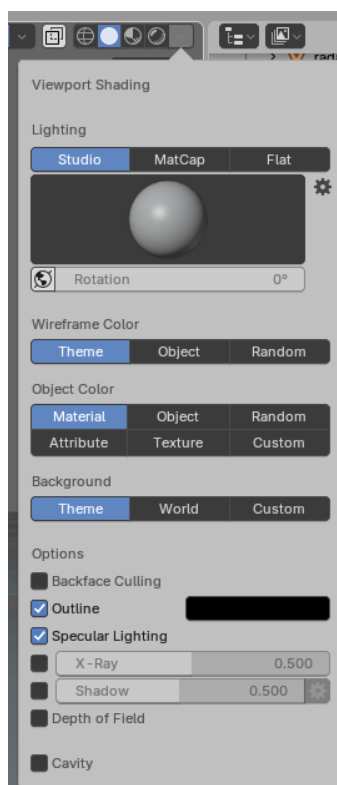


Obrázek 1: Layout

Zdroj: vlastní práce

1.3.3.1 3D Viewport (Hlavní pracovní prostor)

3D Viewport je centrální část záložky Layout, kde uživatel pracuje přímo s objekty ve scéně. Umožňuje manipulaci s objekty, jako je přesun, rotace nebo škálování, a nabízí různé režimy zobrazení, tyto režimy zobrazení mají každé méně či více další možnosti manipulace s objektem. Wireframe (Drátěný model) zobrazuje pouze hrany objektů, což je užitečné pro přesnou práci s geometrií. Solid (Plný model) zobrazuje objekty s jejich základním tvarem a barvou, bez materiálů a textur. Material Preview (Náhled materiálů) zobrazuje objekty s přiřazenými materiály a texturami. Rendered (Renderovaný pohled) zobrazuje objekty tak, jak budou vypadat při konečném renderu, včetně osvětlení a efektů. X-ray (průhledný pohled) je stejný jako Plný model jen s tím rozdílem že skrze objekty lze vidět.

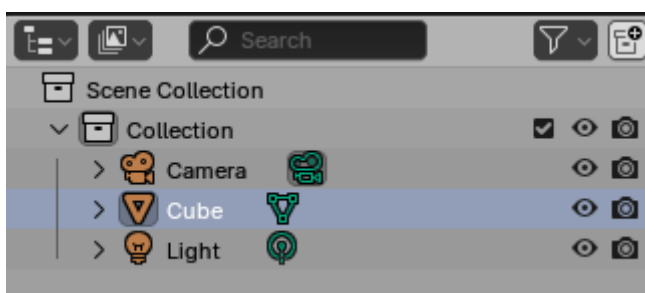


Obrázek 2: View preference

Zdroj: vlastní práce

1.3.3.2 Outliner (Seznam objektů)

Panel v pravé horní části obrazovky, který zobrazuje hierarchii všech objektů ve scéně. Outliner umožňuje snadnou navigaci mezi objekty. Skrývání nebo zamykání objektů, což znemožní s objektem, jakkoliv manipulovat, to slouží pro zjednodušení práce, aby uživatel omylem nepřepínal na jiné objekty. A také tvorbu kolekcí, které slouží k lepší organizaci scény.

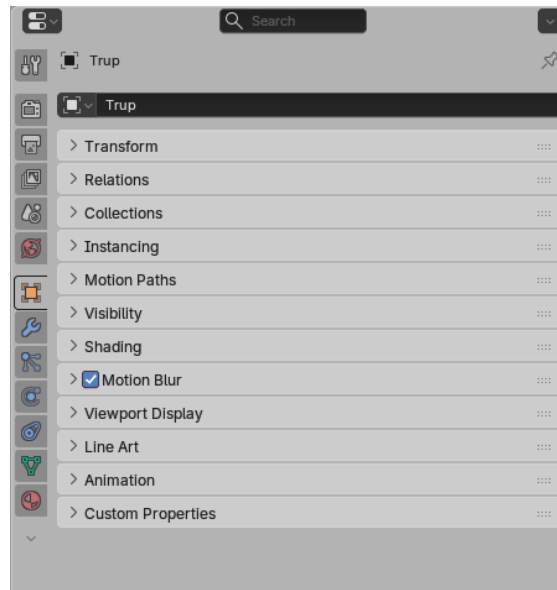


Obrázek 3: Outliner

Zdroj: vlastní práce

1.3.3.3 Properties Editor (Editor vlastností)

Nachází se v pravé dolní části rozhraní. Tento panel umožňuje přístup k podrobným nastavením objektů, jako jsou transformace (poloha, rotace, měřítko), materiály, textury a shaderové nastavení, fyzikální simulace, jako je gravitace, kolize nebo simulace tekutin, nastavení světel a kamer.

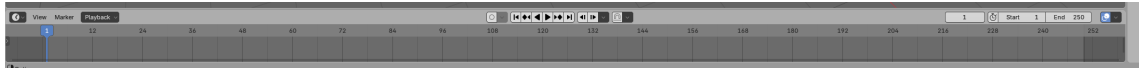


Obrázek 4: Properties editor

Zdroj: vlastní práce

1.3.3.4 Timeline (Časová osa)

Umístěna ve spodní části obrazovky, časová osa je klíčovým nástrojem pro práci s animacemi. Umožňuje nastavovat klíčové snímky, kontrolovat průběh animace, upravit časování nebo rychlost pohybu objektů ve scéně.



Obrázek 5: Timeline

Zdroj: vlastní práce

1.3.3.5 Tool Shelf (Nástrojová lišta)

Panel umístěný vlevo od Viewportu, který obsahuje nejčastěji používané nástroje, jako je, přesouvání objektů ve scéně, otočení objektů kolem os, změna velikosti objektů.



Obrázek 6: Tool shelf

Zdroj: vlastní práce

1.3.4 Klíčové funkce záložky Layout

Záložka Layout nabízí širokou škálu funkcí, které zajišťují efektivní správu a manipulaci se scénou:

Přizpůsobení pracovního prostoru, panely lze přetahovat, dělit, spojovat a různě měnit podle potřeby, třeba pokud místo Properties chcete konzoli na Python apod., což umožňuje uživateli přizpůsobit rozhraní podle vlastních potřeb. Tímto způsobem lze vytvořit optimalizované prostředí pro různé typy projektů.

General	Animation	Scripting	Data
3D Viewport Shift F5	Dope Sheet Shift F12	Text Editor Shift F11	Outliner Shift F9
Image Editor Shift F10	Timeline Shift F12	Python Console Shift F4	Properties Shift F7
UV Editor Shift F10	Graph Editor Shift F6	Info	File Browser Shift F1
Compositor Shift F3	Drivers Shift F6		Asset Browser Shift F1
Texture Node Editor Shift F3	Nonlinear Animation		Spreadsheet
Geometry Node Editor Shift F3			Preferences
Shader Editor Shift F3			
Video Sequencer Shift F8			
Movie Clip Editor Shift F2			

Obrázek 7: Přizpůsobení hlavního prostoru

Zdroj: vlastní práce

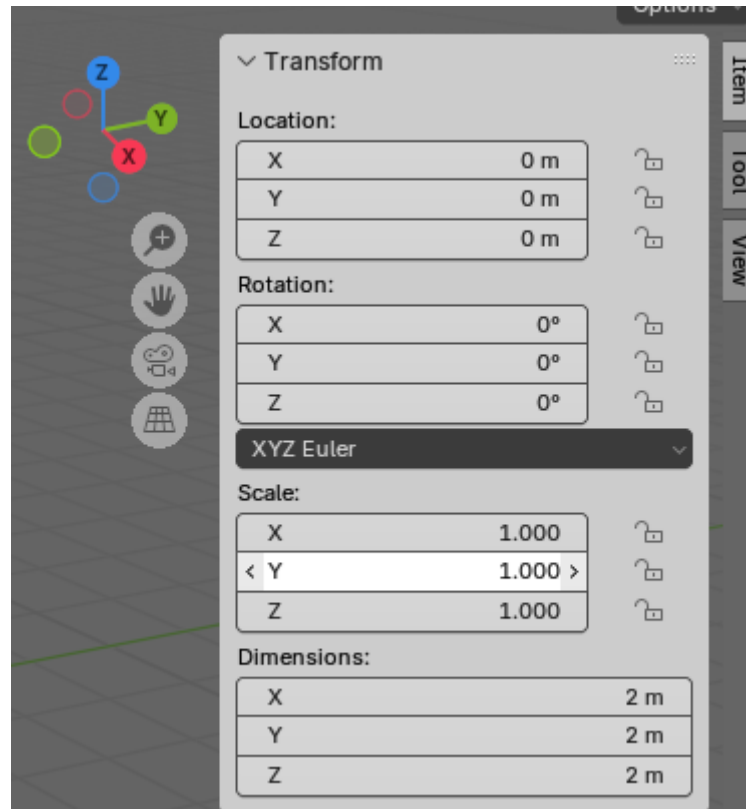
Navigace ve Viewportu, navigace ve 3D prostoru je klíčová. Pomocí myši a klávesových zkratk může mimo jiné uživatel otočit pohled, přiblížit nebo oddálit scénu, posouvat pohled

Práce s kamerou

Kamera je základním prvkem pro finální vizualizaci scény. V záložce Layout lze kameru snadno nastavit, přesouvat a upravovat její záběr. Klávesová zkratka *O* na numerické klávesnici umožňuje přepnout pohled přímo do kamery.

Editace objektů

Manipulace s objekty je možná nejen ručně, ale také numericky prostřednictvím panelu Transform v pravém horním rohu Viewportu.



Obrázek 8: Transform panel

Zdroj: vlastní práce

Nastavení osvětlení a stínů

Záložka Layout umožňuje rychlou správu světel ve scéně. Uživatel může přidávat různé typy světel (např. bodové, směrové nebo plošné) a sledovat jejich vliv na scénu v reálném čase.

Záložka Layout je všestranným a nezbytným nástrojem v Blenderu. Je navržena tak, aby usnadnila organizaci a správu scény a poskytla všechny základní funkce na jednom místě. Díky své přizpůsobitelnosti a uživatelské přívětivosti je ideální jak pro začátečníky, tak pro pokročilé uživatele. Záložka Layout je výchozím bodem každého projektu a hraje klíčovou roli při přípravě na další fáze práce, jako je modelování, texturování nebo animace. (Blender, 2024)

1.3.5 Modeling

Záložka Modeling v Blenderu je specializovaným pracovním prostředím určeným pro tvorbu a úpravu 3D modelů. Umožňuje detailní práci s geometrií objektů a poskytuje nástroje pro tvorbu komplexních tvarů, úpravy polygonů, a manipulaci s okraji, plochami a vrcholy. Záložka Modeling je základním bodem pro modelování a design ve 3D prostoru.

1.4 Autodesk 3DS MAX

Autodesk 3DS Max je profesionální program pro 3D grafiku, vizualizaci a animaci. Bývá používán v postprodukci, při výrobě reklam, filmů a v televizním průmyslu obecně, pro architektonické a konstrukční vizualizace a často slouží i k tvorbě grafiky do počítačových her.

1.4.1 Historie

3D Studio Max, později známé jako 3ds Max, je jedním z nejstarších a nejvýznamnějších nástrojů pro 3D modelování a animaci. Vznik tohoto softwaru lze vystopovat do 80. let, kdy Gary Yost, digitální umělec a vizionář, spolu se svým týmem začal vyvíjet první verzi softwaru pro DOS. V roce 1990 Yost Group navázala partnerství s Autodeskem, který odstranil omezení paměti a přidal animaci. Autodesk komerčně uvedl 3D Studio za cenu 3495 dolarů, což bylo v té době na plně vybavený 3D software relativně levné.

První verze 3D Studio byla vydána v 80. letech a byla určena pro operační systém DOS. Software vyžadoval pouhých 640 kb RAM, což bylo v té době technologickým limitem, ale zároveň důkazem důmyslnosti jeho designu. V roce 1990 Yost Group, vedená Garym Yostem, navázala spolupráci s firmou Autodesk, která se stala vydavatelem softwaru. Díky takovéto spolupráci byla odstraněna omezení paměti, přidána animace a software byl uveden na trh za cenu 3495 USD. Částka byla na plně vybavený 3D nástroj poměrně nízká, což 3D Studio odlišovalo od konkurence.

V letech 1992–1994 byly do softwaru přidány pluginy, což umožnilo vývojářům a uživatelům vytvářet vlastní rozšíření. Mezi další inovace patřila inverzní kinematika, rychlý náhled, skriptování klíčových snímků a předvolby pro částicové systémy. Novinky umožnily použití 3D Studio v profesionálních projektech.

Roky 1995–1996 přinesly zásadní změnu s přechodem na Windows NT, což otevřelo nové možnosti díky 32bitové platformě. Software byl přejmenován na 3D Studio MAX a nabídl funkce jako animace v reálném čase a nástroje pro modelování postav. Mezi ikonické ukázky jeho možností patří „Dancing Baby“, jedno z prvních internetových memů.

V roce 1997 Autodesk přidal MAXScript, což umožnilo uživatelům vytvářet vlastní nástroje a automatizovat pracovní postupy. V roce 1999 Autodesk koupil společnost Discreet Logic a technologie obou firem byly integrovány do softwaru.

V roce 2000 proběhl rebranding na 3DS max.

V roce 2006 byl software přejmenován na Autodesk 3ds Max, pod kterým je znám dodnes. Změny v označování verzí, přechod na názvy dle roků vydání, začaly v roce 2007. 3ds Max se od té doby nadále vyvíjí a zůstává jedním z nejdůležitějších nástrojů v oblasti 3D modelování, animace a vizuálních efektů. (GarageFarm.NET, 2024)

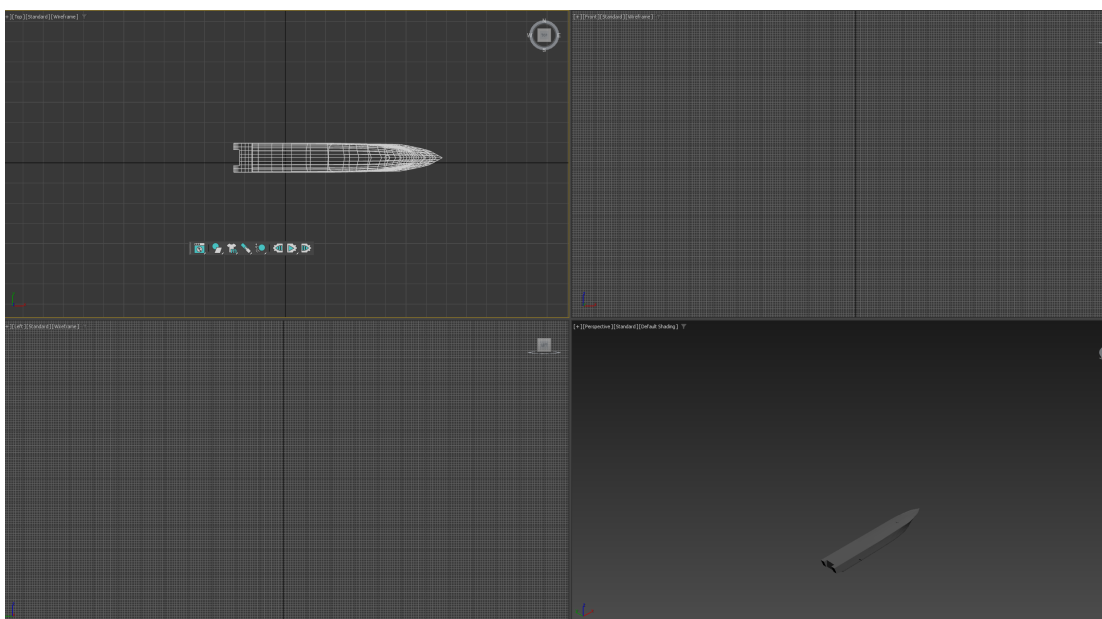
1.4.2 Nástroje a funkce

Podobně jako Blender, i Autodesk 3DS Max disponuje širokou paletou nástrojů pro tvorbu komplexní 3D grafiky. Jeho filozofie uživatelského rozhraní se však historicky vyvíjela odlišně a klade velký důraz na centralizované ovládání prostřednictvím takzvaného Command Panelu a vrstvení modifikátorů.

1.4.3 Pracovní prostory a Viewporty

Zatímco Blender standardně otevírá jeden velký 3D Viewport, 3DS Max je tradičně rozvržen do čtyř pracovních oken (Viewportů) současně. Standardní rozložení zahrnuje pohled shora (Top), zepředu (Front), zleva (Left) a perspektivní pohled (Perspective). Toto rozvržení usnadňuje

precizní umístění objektů ve 3D prostoru, nicméně uživatel si může rozložení libovolně upravit nebo maximalizovat pouze jedno okno. Podobně jako u jiných programů lze přepínat mezi různými styly zobrazení, jako je drátěný model (Wireframe), plný model (Default Shading) nebo zobrazení s hrubými hranami (Edged Faces).



Obrázek 9: Viewporty

Zdroj: vlastní práce

1.4.4 Comand Panel

Command Panel se nachází v pravé části obrazovky a je absolutním srdcem celého programu 3DS Max. Obsahuje téměř všechny nástroje potřebné pro tvorbu a úpravu scény. Je rozdělen do šesti hlavních karet:

Create (Vytvořit): Slouží k přidávání nových objektů do scény, ať už jde o 3D geometrii (Standard Primitives), 2D tvary (Splines), světla, kamery nebo pomocné objekty.

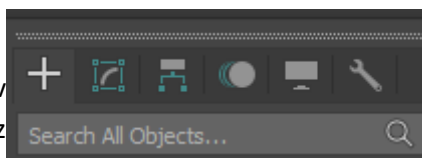
Modify (Upravit): Karta určená pro úpravu parametrů již existujících objektů. Zde se nachází klíčový nástroj programu – Modifier Stack.

Hierarchy (Hierarchie): Správa propojení objektů (Parent/Child) a nastavování jejich středových bodů (Pivotů).

Motion, Display a Utilities: Karty pro pokročilou správu animací, zobrazení objektů a doplňkové nástroje.

1.4.5 Modifier Stack

Modifier Stack, umístěný v kartě Modify, je jedním z nejmocnějších a



Obrázek 10: Command panel

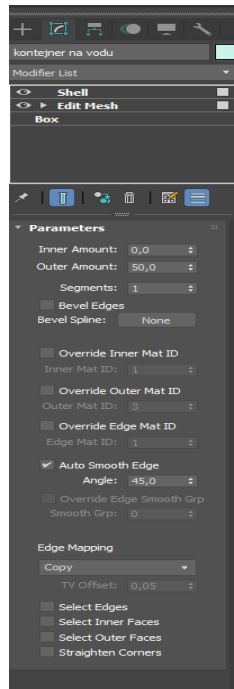
Zdroj: vlastní práce

nejcharakterističtějších

nástrojů 3DS Max. Jedná se o

seznam aplikovaných úprav (modifikátorů) na vybraný objekt. Modifikátory se na sebe vrství

destruktivním i nedestruktivním způsobem. Uživatel může kdykoliv v zásobníku přejít na nižší úroveň, upravit základní tvar objektu, a změna se automaticky propíše přes všechny nadřazené modifikátory. Modifikátory lze také libovolně vypínat, měnit jejich pořadí nebo celou historii úprav takzvaně „zapéct“ (Collapse) do čisté geometrie pro ušetření výkonu počítače. (Autodesk, 2024)

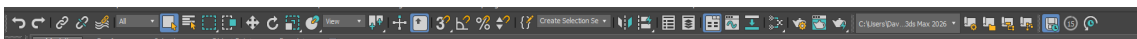


Obrázek 11: Modifier stack

Zdroj: vlastní práce

1.4.6 Main Toolbar

Hlavní panel se nachází v horní části obrazovky pod hlavní nabídkou. Obsahuje ikony pro nejčastěji používané akce, jako je výběr objektů (Select Object), základní transformace (Move, Rotate, Scale), nastavení přichytávání (Snaps) pro přesnou práci s geometrií, a také rychlý přístup



Obrázek 12: Main toolbar

Zdroj: vlastní práce

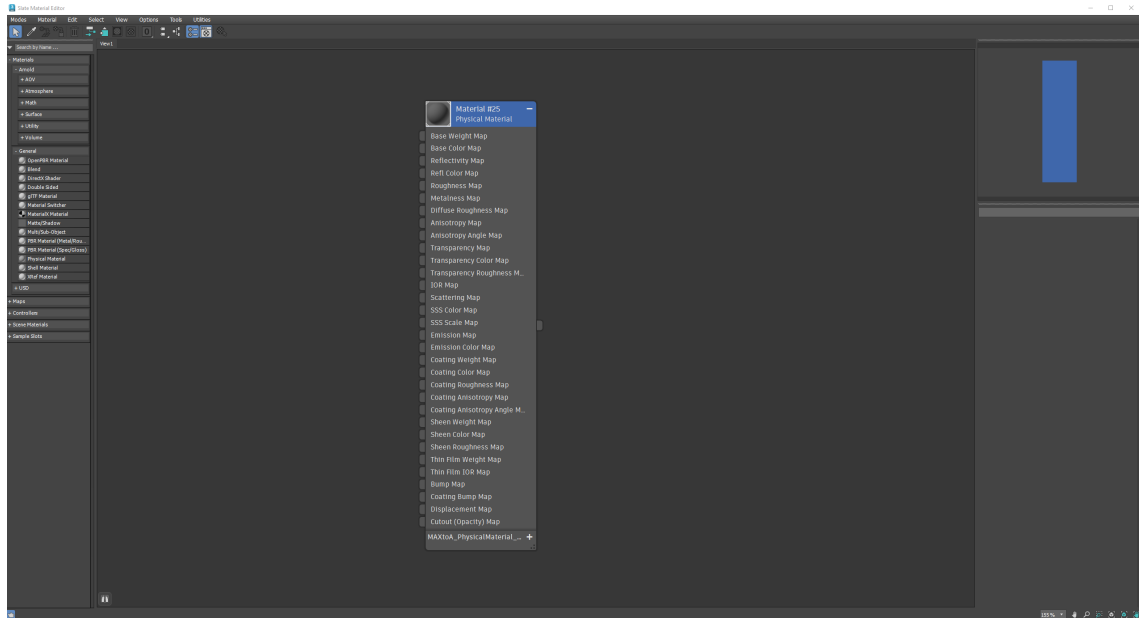
k editoru materiálů a nastavení renderování.

1.4.7 Time Slider a Track Bar

Ve spodní části rozhraní se nachází nástroje pro animaci. Time Slider slouží k pohybu v čase (mezi jednotlivými snímky animace), zatímco Track Bar pod ním vizuálně zobrazuje klíčové snímky (Keyframes) pro vybraný objekt. Systém animace úzce spolupracuje s funkcí Auto Key, která automaticky zaznamenává jakékoliv změny parametrů objektu v čase.

1.4.8 Material Editor

3DS Max nabízí dva režimy pro tvorbu materiálů. Compact Material Editor, který se používá spíše historicky pro rychlé a jednoduché textury, a moderní Slate Material Editor. Ten využívá systém propojování jednotlivých uzlů (Node-based workflow), který umožňuje vytvářet vizuálně velmi komplexní materiály sítěmi textur, což je systém velmi podobný záložce Shading v Blenderu.



Obrázek 13: Material editor

Zdroj: vlastní práce

2 Praktická část

V praktické části této bakalářské práce je realizováno srovnání programů Blender a Autodesk 3DS Max na reálném příkladu tvorby a simulace 3D modelu bitevní lodi. Cílem je porovnat workflow, efektivitu nástrojů a schopnost obou programů pracovat s fyzikálními simulacemi ve vodním prostředí.

2.1 Výběr předlohy a metodika modelování

Pro účely srovnání byla zvolena americká bitevní loď třídy Littoral Combat Ship, konkrétně varianta LCS-1 (třída Freedom). Tato volba byla učiněna na základě strategického rozhodnutí omezit geometrickou složitost trupu. Oproti variantě LCS-2 (třída Independence), která využívá trimaranovou konstrukci, disponuje LCS-1 čistším monohull designem. Tato geometrie umožňuje objektivněji porovnat základní nástroje pro práci s topologií v obou programech, aniž by srovnání bylo zkresleno extrémně složitými přechody mezi plováky.

Jako hlavní metodika byla zvolena technika Subdivision Surface (SubD) modelování. Tento postup spočívá v tvorbě zjednodušené základní sítě (tzv. blockout), na kterou je následně aplikován vyhlazovací algoritmus. (Vaughan, 2012)

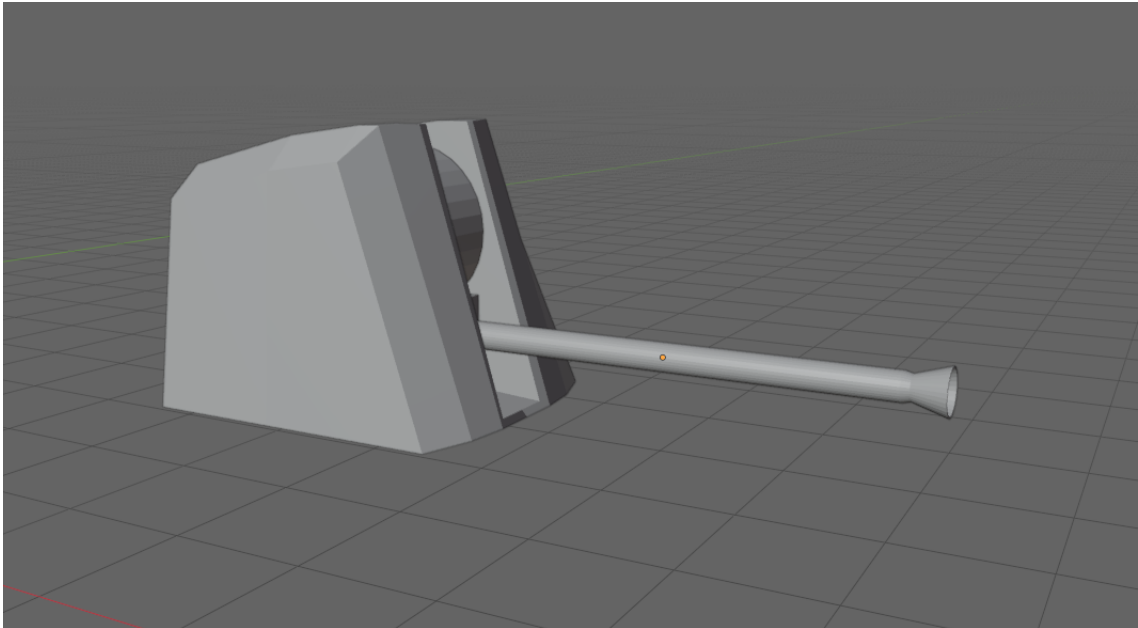
2.2 Tvorba modelu v Blenderu

Pro zajištění maximální tvarové věrnosti a dodržení správných proporcí byl jako referenční podklad využit technický blueprint lodi třídy Freedom (LCS-1). Tento výkres představoval exaktní šablonu pro definování délky trupu, sklonu přídě a rozmístění jednotlivých prvků nadstavby. Zatímco v prostředí 3ds Max vyžaduje aplikace referencí mapování textur na pomocné roviny (Plane) v ortografických pohledech, Blender umožňuje vložení referenčního obrázku (Background Image) přímo do 3D viewportu. Tento metodický krok byl zcela klíčový pro zachování měřítka a prostorové konzistence mezi oběma zkoumanými softwary.

2.2.1 Modelování

Samotný proces modelování byl zahájen v pracovním prostoru *Layout* vytvořením základní geometrie typu *Cube*. Ta byla v režimu *Edit Mode* postupně tvarována pomocí standardních polygonálních nástrojů, primárně funkcí *Extrude* (vytahování ploch) a *Loop Cut* (přidávání topologických řezů).

Pro dosažení plynulého hydrodynamického tvaru trupu byl aplikován modifikátor *Subdivision Surface*. V této fázi byl kladen striktní důraz na udržení čisté topologie tvořené výhradně čtyřúhelníky (*Quads*). Tímto přístupem se proaktivně zamezilo vzniku nežádoucích vizuálních artefaktů a stínových deformací, které často vznikají při aplikaci vyhlazovacích algoritmů na víceúhelníky (*N-gony*). K zachování definovaných technických hran, zejména na zádi lodi, byla využita funkce *Mean Crease*, která modifikátoru explicitně určuje hrany s nulovým stupněm vyhlazení.



Obrázek 14: Dělo_Blender

Zdroj: vlastní práce

2.2.2 Detailizace

Finalizace modelu lodi LCS-1 probíhala s využitím technik zaměřených na efektivitu a minimalizaci zbytečného navyšování hustoty polygonální sítě:

Využití symetrie: Klíčovým nástrojem pro zachování souměrnosti byl modifikátor Mirror. Jeho aplikace na většinu komponent (trup, nadstavba, doplňky) zajistila absolutní osovou přesnost plavidla a zredukovala časovou náročnost modelovací fáze na polovinu.

Tvorba radarů a doplňků: Konstrukce sensorových dómů a komunikačních prvků vycházela z topologie základních primitiv (válec, koule). Pomocí metody vytahování (*Extrude*) byly tyto tvary postupně modifikovány do proporcí odpovídajících technické předloze.

Zkosení hran: Pro zjemnění ostrých přechodů a dosažení přirozenějšího lomu světla byl využit modifikátor Bevel. Tento nástroj se ukázal jako stěžejní zejména při konstrukci hlavního stožáru, kde umožnil vizuální přiblížení se předloze bez nutnosti destruktivních zásahů do základní geometrie.

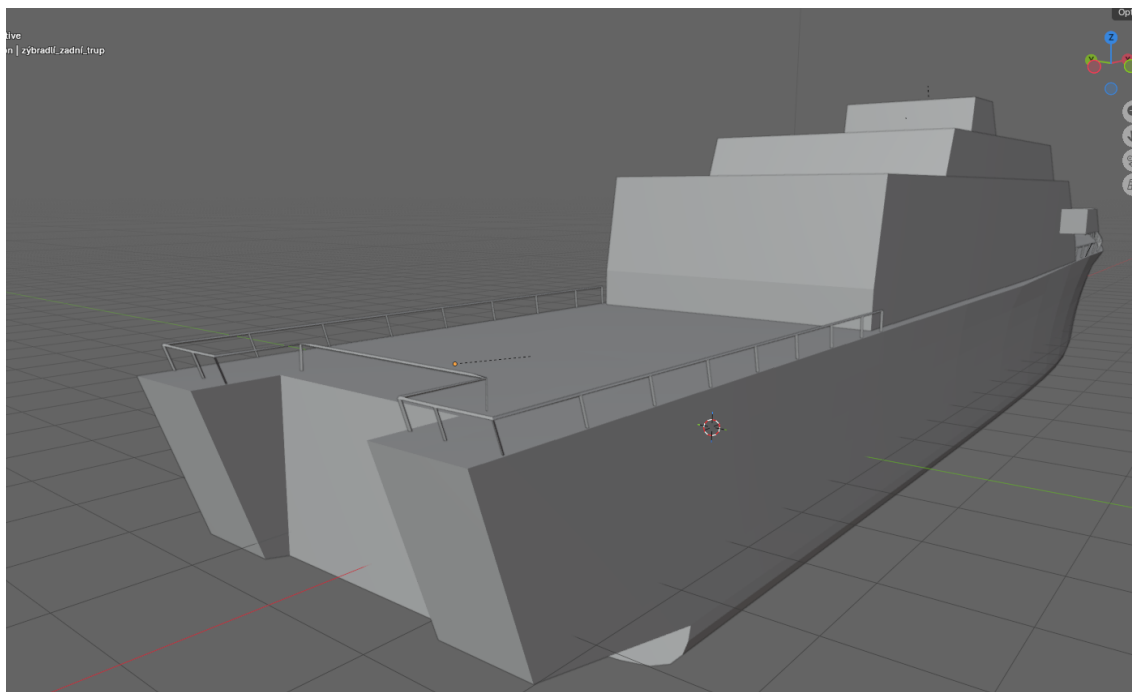
Logické operace: Modifikátor Boolean byl nasazen lokálně pro úpravu specifických detailů, u kterých by klasické manuální prořezávání sítě bylo neefektivní. Jednalo se především o modelování palubních výřezů a lůžek pro zadní dělové věže.

Okna můstku: Zasklení a rámy oken byly vytvořeny metodou separace geometrie. Vybrané polygony nadstavby byly zduplikovány, odděleny do samostatného objektu a následně pomocí nástroje *Extrude* vytaženy podél normál směrem dovnitř, čímž vznikla požadovaná hloubka bez narušení integrity hlavní sítě trupu.

2.2.3 Generování bezpečnostního zábradlí

Při tvorbě bezpečnostního zábradlí podél paluby byla zvolena vysoce efektivní metoda nedestruktivního modelování pomocí modifikátoru Wireframe. Postup spočíval v duplikaci

obvodových hran geometrie trupu a jejich následném oddělení do samostatného objektu. Na tento 1D profil byl aplikován zmíněný modifikátor, který automaticky transformoval pouhé hrany sítě na trojrozměrnou prutovou konstrukci o definované tloušťce. Tato technika zajistila, že zábradlí zcela exaktně kopíruje zakřivení a sklon boků lodi, což představuje výrazně přesnější a rychlejší řešení v porovnání s ručním umístováním jednotlivých sloupků.



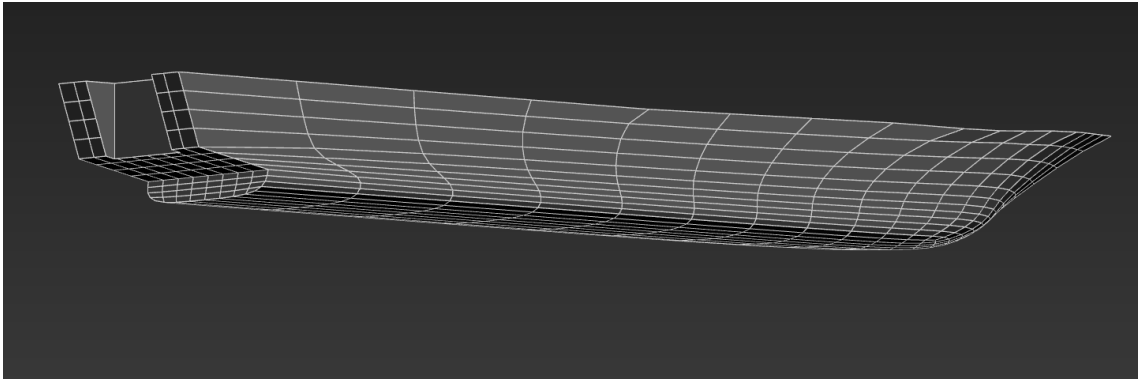
Obrázek 15: Zád lodi_ Blender

Zdroj: vlastní práce

2.3 Tvorba modelu v 3DS MAX

Při práci v Autodesk 3ds Max byl zachován identický postup jako v prostředí Blender, aby bylo možné objektivně porovnat nativní logiku obou nástrojů. Přestože moderní výměnné formáty, jako je FBX, umožňují přenos geometrie mezi aplikacemi, v rámci analýzy byla geometrie trupu testována přímo v prostředí 3ds Max. Důvodem je skutečnost, že při importu dochází ke ztrátě procedurální historie modelu, zejména modifikátorového zásobníku, což omezuje možnosti následné nedestruktivní editace.

V průběhu modelování byly identifikovány zásadní rozdíly ve workflow, terminologii a logice ovládání, které vyžadovaly odlišný technický přístup k řešení geometrie trupu, a také ovlivnily rychlost i technické provedení jednotlivých prvků lodi:



Obrázek 16: Trup lodi v 3DS MAX

Zdroj: vlastní práce

2.3.1 Symetrie a konstrukce trupu

Při konstrukci symetrického trupu lodi došlo k identifikaci zásadního rozdílu v názvosloví nástrojů. Na základě zkušeností z programu Blender, kde je pro zrcadlení geometrie využíván modifikátor *Mirror*, byl v 3ds Max nejprve intuitivně zvolen nástroj Mirror z horní nástrojové lišty. Následným testováním bylo zjištěno, že tento nástroj pracuje na úrovni objektové transformace, což vedlo k vytvoření dvou separátních polovin trupu bez vzájemného propojení vertexů. Tento postup se ukázal jako nepoužitelný pro následné vyhlazování (*Subdivision*), neboť ve středové ose vznikal viditelný šev.

Řešením byla aplikace modifikátoru *Symmetry*. Na rozdíl od základního zrcadlení pracuje *Symmetry* na úrovni geometrie, kde provádí automatický ořez přesahujících ploch (*Slice*) a především kriticky důležité svaření vrcholů (*Weld*) v ose symetrie. Tento poznatek zdůrazňuje, že ačkoliv oba programy nabízejí podobné funkce, jejich implementace a názvosloví se natolik liší, že vyžadují precizní recalibraci workflow při přechodu mezi nimi.

2.3.2 Subdivize a vyhlazování hran

Samotné modelování probíhalo metodou *Box Modelingu* s následnou konverzí na Editable Poly.

Členění sítě: Pro detailizaci nadstavby byl využíván nástroj *Swift Loop* z horního Ribbon menu. Oproti Blenderu, kde je rozdělení ploch řešeno interaktivně přes klávesové zkratky, vyžaduje 3ds Max častější interakci s grafickým rozhraním panelů.

Subdivize: Pro zachování ostrých technických hran nadstavby byl zvolen pokročilý modifikátor *OpenSubdiv*. Standardní modifikátor *TurboSmooth* byl vyhodnocen jako nedostačující, neboť na rozdíl od *Subdivision Surface* v Blenderu nedokáže nativně pracovat s váhami hran (*Edge Crease*). *OpenSubdiv* umožnil definovat ostrost hran bez nutnosti přidávání nadbytečné podpůrné geometrie (*support loops*).

2.3.3 Konstrukce zábradlí

Stejně jako v programu Blender, i v prostředí 3ds Max byla tvorba bezpečnostního zábradlí odvozena z existující topologie trupu, aby byla zajištěna tvarová návaznost na palubu. Vzhledem

k odlišné logice obou programů se však postup z hlediska nástrojů lišil a vyžadoval kombinaci technik.

Zatímco Blender efektivně využívá plošný modifikátor *Wireframe*, v 3ds Max byla zvolena metoda kombinující křivky (*Splines*) a parametrickou distribuci objektů. Postup byl rozdělen na dvě fáze:

Horizontální madla: V režimu úprav hran (*Edge*) byly extrahovány obvodové hrany paluby pomocí funkce *Create Shape from Selection*. Vzniklá křivka byla následně definována jako renderovatelný objekt s kruhovým průřezem aktivací voleb *Enable in Renderer/Viewport*. Tím byla vytvořena souvislá horizontální konstrukce zábradlí.

Vertikální sloupky: Pro vytvoření sloupků byl vymodelován jeden vzorový válec. Ten byl následně distribuován podél dříve vytvořené křivky (madla) pomocí nástroje *Spacing Tool*, který umožnil přesné nastavení počtu a rovnoměrných rozestupů sloupků, jež byly následně automaticky svařeny (*Weld*) do jednoho objektu s madlem.

Tato kombinace technik se ukázala jako efektivnější než využití přímého ekvivalentu nástroje *Wireframe* (modifikátor *Lattice*), neboť poskytuje preciznější kontrolu nad tloušťkou jednotlivých prvků a plynulostí oblouků bez vzniku nežádoucí geometrie ve spojích.

2.4 Definice povrchových vlastností a PBR materiálů

Nedílnou součástí produkčního workflow v obou programech bylo vytvoření odpovídajících povrchových vlastností (shading) plavidla i radarových dómů. Ačkoliv se uživatelské rozhraní testovaných softwarů vizuálně liší, oba plně podporují průmyslový standard fyzikálně korektního renderování (PBR – Physically Based Rendering). Tento standard zaručuje konzistentní chování světla na povrchu modelu napříč různými renderovacími enginey. (McDermott, 2018)

2.4.1 Nativní shading a real-time odezva v softwaru Blender

Pro definici materiálů v Blenderu byl využit integrovaný *Shader Editor*, který funguje na bázi vizuálního programování pomocí uzlů (Node-based workflow). Základním stavebním kamenem všech povrchů se stal univerzální uzel *Principled BSDF*, do kterého byly zapojeny příslušné texturové mapy (Base Color, Metallic, Roughness a Normal).

Zásadní komparativní výhodou prostředí Blenderu se v této fázi ukázala integrace real-time renderovacího engineu *Eevee*. Ten umožňuje okamžitou vizuální zpětnou vazbu přímo ve Viewportu bez nutnosti zdlouhavého předpočítávání scény. Uživatel tak může v reálném čase ladit odlesky či hrubost povrchu, což drasticky snižuje časovou náročnost iteračního procesu před finálním renderem v raytracingovém engineu *Cycles*.

2.4.2 Architektura Slate Material Editoru v 3DS MAX

V softwaru 3ds Max byla tvorba materiálů realizována prostřednictvím modulu *Slate Material Editor*, který rovněž nabízí nodovou architekturu. Jako funkční ekvivalent k řešení v Blenderu byl použit nativní *Physical Material*, jenž je plně optimalizován pro integrovaný renderovací engine *Arnold*. Z hlediska propojování PBR textur je technologická logika identická s Blenderem.

Rozdíl se však projevilo v celkové plynulosti pracovního postupu. 3ds Max vyžaduje pro korektní projekci textur častější aplikaci a manuální ladění modifikátoru *UVW Map*. Náhled komplexnějších PBR materiálů přímo v pracovním okně (ActiveShade) je navíc zřetelně pomalejší a méně responzivní než engine Eevee v Blenderu. Proces nastavování materiálů si tak v 3ds Max vyžádal vyšší počet testovacích výpočtů (IPR – Interactive Photorealistic Rendering), což celou produkční fázi shadingu úměrně zpomalilo.

2.5 Simulace vodního prostředí a interakce plavidla v Blenderu

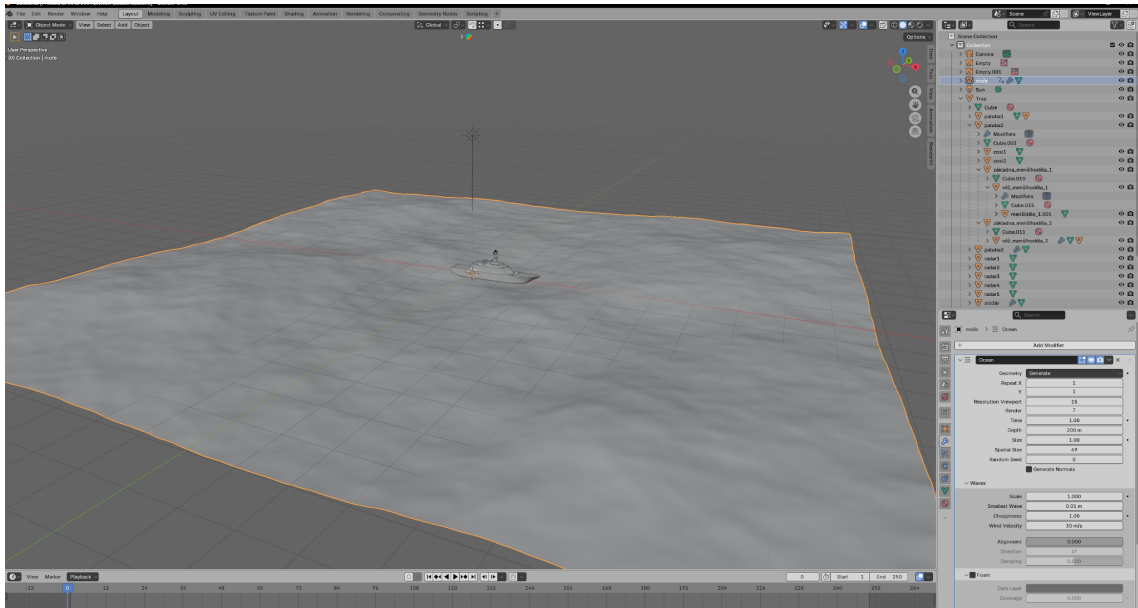
Pro komplexní demonstraci schopností programu Blender v oblasti simulace okolní vody a plnění zadání práce byl zvolen tzv. hybridní přístup. Program disponuje integrovaným systémem Mantaflow, který počítá reálnou objemovou dynamiku tekutin na základě Navier-Stokesových rovnic. Aplikace tohoto plnohodnotného částicového solveru na celou rozlohu širého oceánu by však dalece přesáhla výpočetní kapacity (paměť RAM a strojový čas CPU/GPU) běžných pracovních stanic a znemožnila by iterativní ladění scény.

Z tohoto důvodu byl fyzikální kapalinový simulátor Mantaflow využit přísně lokálně – formou omezené simulační domény (Local Bounding Box). Tato doména byla umístěna výhradně do míst s předpokládanou dramatickou fyzikální interakcí, konkrétně pro výpočet reálné vodní tříště a cákanců při dopadu fragmentů zničené nadstavby do moře. Pro generování zbytku rozlehlé vodní plochy a simulaci běžné interakce plavidla s hladinou byl následně nasazen hardwarově efektivnější procedurální přístup.

2.5.1 Vliv topologie na stabilitu logických operací

Základ okolní vodní plochy byl vytvořen pomocí nedestruktivního modifikátoru Ocean. Tento nástroj negeneruje fyzikální částice, nýbrž matematicky deformuje polygonální síť (Mesh) na základě algoritmu FFT (Fast Fourier Transform), který přímo vychází z reálných oceánografických modelů vlnění.

Tento postup umožnil vytvořit vizuálně přesvědčivou hladinu sahající až k horizontu. Dynamika moře byla animována pomocí klíčování parametru času (*Time*), což zajistilo plynulé přelévání vln. Úroveň detailů sítě (*Resolution*) a celková výška vln (*Scale*) byly optimalizovány v proporcích k vloženému plavidlu třídy LCS-1. Výpočet této hladiny probíhá v reálném čase, čímž zcela odpadá nutnost zdlouhavého předpočítávání (Baking) obrovských cache souborů.



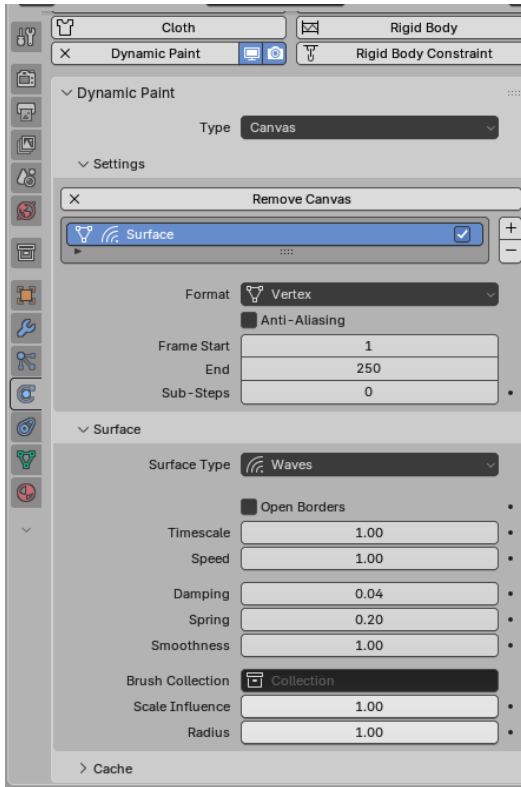
Obrázek 17: "Falešný" oceán

Zdroj: vlastní práce

2.5.2 Interakce plavidla a procedurální hladiny systémem Dynamic Paint

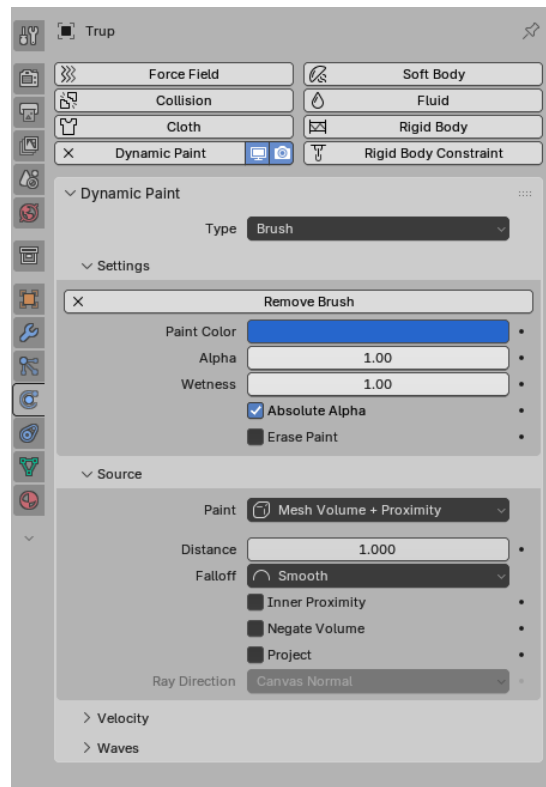
Pro zajištění reakce procedurální hladiny na plující plavidlo – tvorbu příďové vlny a brázdy (Wake) – byl nasazen systém Dynamic Paint (Dynamické malování).

V rámci tohoto engine byl procedurální oceán definován jako reaktivní plátno (*Canvas*) s vlastností generování vln (*Waves*). Trup plavidla fungoval jako kolizní štětec (*Brush*). Pro přesnou reakci vody na vytlačovaný objem lodi byl zdroj štětce nastaven na kombinaci objemu a vzdálenosti (*Mesh Volume + Proximity*). Po animování dopředného pohybu lodi začal systém automaticky deformovat geometrii oceánu a tvořit za plavidlem realisticky se šířící brázdu, jejíž parametry útlumu (*Damping*) byly manuálně doladěny. Tento hybridní postup prokazuje, že Blender dokáže poskytnout plnohodnotný vizuální i fyzikální výsledek kapalinové simulace při zlomkové spotřebě systémových prostředků.



Obrázek 18: Dynamic paint_canvas

Zdroj: vlastní práce



Obrázek 19: Dynamic paint_brush

Zdroj: vlastní práce

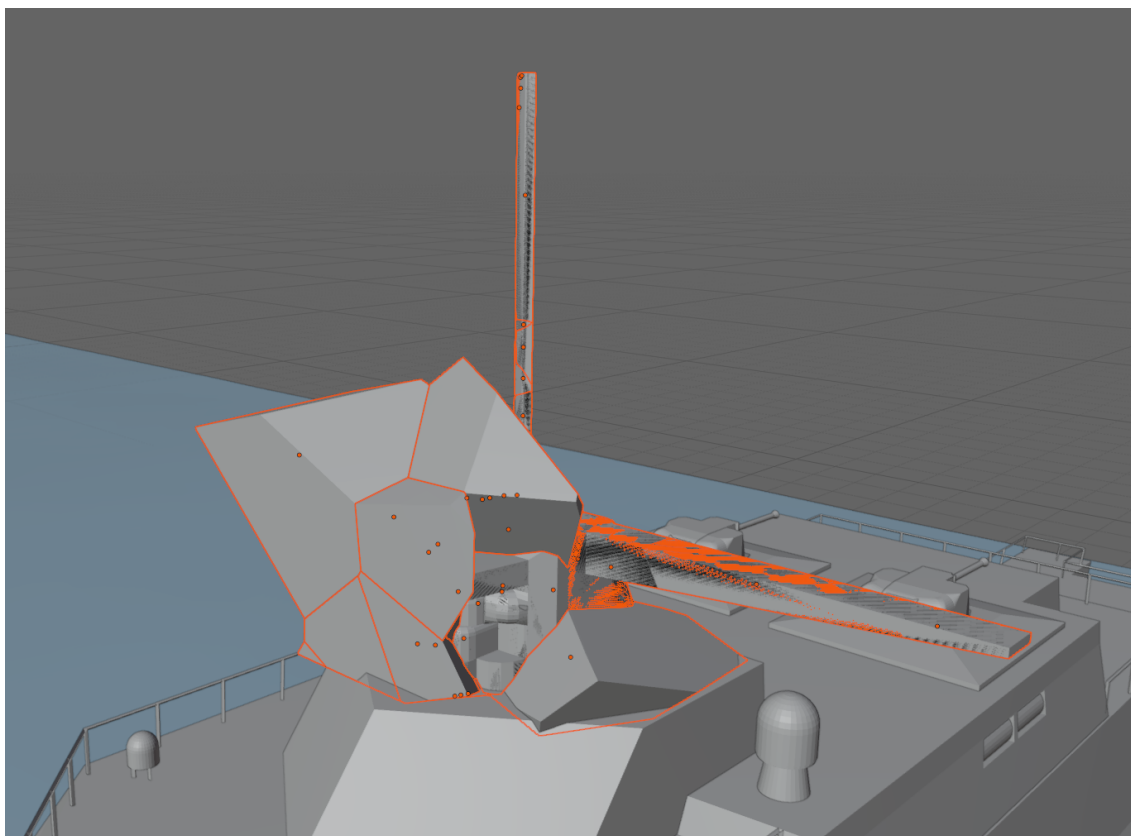
2.6 Destrukce v Blenderu

V rámci komplexního posouzení simulačních schopností programu Blender byl realizován test fyzikální destrukce rigidních těles (Rigid Body Dynamics).

2.6.1 Limity nástroje Cell Fracture při práci s detailní topologií

Během úvodního testování procedurální fragmentace (prostřednictvím integrovaného nástroje *Cell Fracture*) na modelu hlavního komunikačního stožáru byla odhalena zásadní slabina algoritmu při práci s komplexní topologií. Bylo zjištěno, že objekty, na které byl v průběhu modelování aplikován modifikátor či plošný nástroj Bevel (zkosení hran), vykazují kritickou chybovost při generování úlomků.

Algoritmus Cell Fracture, jenž je výpočetně postaven na logických operacích (Booleans), nedokázal korektně zpracovat mikroskopické plošky a složité mnohoúhelníky (N-gony) vznikající ve vrcholech zkosených hran. Tento technický konflikt vedl ke zhroucení výpočtu objemu a generování zdeformovaných či nedokonalých fragmentů. V rámci optimalizace workflow a zajištění efektivity produkce tak bylo nutné upravit metodiku: pro fyzikální destrukci byly záměrně vybrány alternativní cíle na plavidle, které disponovaly čistší topologií bez mikroskopických detailů.



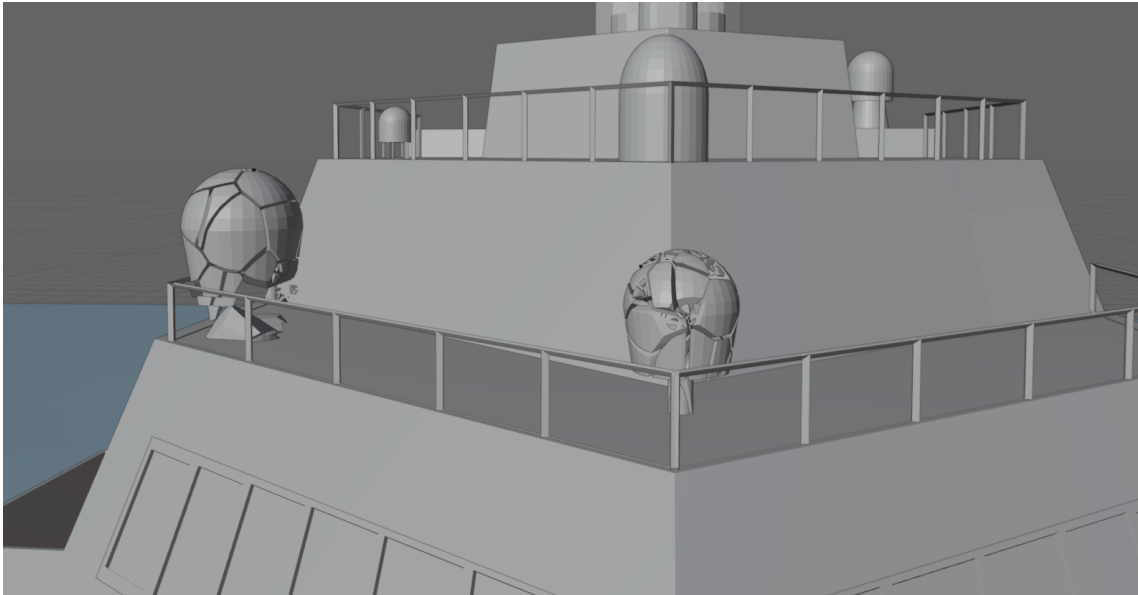
Obrázek 20: Test cell fracture

Zdroj: vlastní práce

2.6.2 Výběr cíle a procedurální fragmentace

Jako ideální cíl pro simulaci střelby byly zvoleny dva radarové dómy umístěné na palubě v lineární ose. Tato konfigurace byla vybrána záměrně pro otestování výpočtu přenosu kinetické energie mezi vícero nezávislými objekty při jediném průletu projektilu (tzv. penetrační zásah).

Rozbití geometrie dómů proběhlo pomocí nástroje *Cell Fracture*. Pro zachování optimálního poměru mezi vizuálním detailem a výpočetní náročností byl parametr *Source Limit* (počet generovaných úlomků) nastaven na hodnotu několika desítek kusů pro každý dóm.



Obrázek 21: Cell fracture

Zdroj: vlastní práce

2.6.3 Fyzikální model a dynamika destrukce

Po úspěšném rozdělení geometrie bylo nezbytné aplikovat fyzikální vlastnosti pro výpočet samotné kolize. Všem vygenerovaným úlomkům byl přiřazen systém dynamiky pevných těles (*Rigid Body*) s typem *Active*. Jako tvar kolizního obalu (*Collision Shape*) byl zvolen *Convex Hull*, což zajišťuje dostatečně přesný výpočet odrazů fragmentů od sebe navzájem i od paluby při zachování optimalizovaného výpočetního výkonu. Samotný trup plavidla byl pro účely simulace definován jako *Passive*, aby plnil roli pevné překážky, po které se mohou trosky realisticky rozptýlit.

Aby se fragmenty nezhroutily vlivem gravitace dříve, než dojde k samotnému zásahu projektilem, byl v dynamice úlomků aktivován parametr *Start Deactivated*. Fyzikální solver tak začne počítat setrvačnost a gravitaci pro konkrétní objekt až v moment, kdy do něj narazí jiné aktivní těleso (projektil).

2.6.4 Kinematický přepínač a alternativní přístupy k destrukci

Během implementace fyzikální simulace na pohybujícím se plavidle vyvstal technický problém s dědičností translačních matic. Fyzikální engine počítá dynamiku primárně v globálním souřadnicovém prostoru (*World Space*), což vyvolávalo značné konflikty při pokusech o standardní hierarchické provázání (*Parenting*) dynamických úlomků s plující lodí. Objekty řízené fyzikou měly tendenci ignorovat pohyb rodičovského objektu a fixovat se na výchozí souřadnice.

Tento problém byl ve finální scéně vyřešen využitím tzv. kinematického přepínače (klíčování vlastnosti *Animated* v modulu *Rigid Body*). Tento postup zajišťuje, že se úlomky až do okamžiku nárazu chovají jako striktně kinematická tělesa plně dědící pohyb lodi. Přesně ve snímku průrazu projektilu je parametr *Animated* programově deaktivován, čímž kontrolu nad tělesy okamžitě přebírá dynamický solver a zahajuje realistický výpočet disperze trosek.

V rámci vývoje a testování scény byla zvažována i alternativní optimalizační technika zvaná *Model Swapping*. Tato metoda spočívá v absolutním skrytí intaktního modelu a jeho bleskové výměně za model fragmentovaný přesně ve snímku zásahu. Přestože se jedná o hojně využívaný postup ve filmové praxi, který elegantně obchází konflikty souřadnicových systémů, pro finální implementaci byl upřednostněn kinematický přepínač. Ten poskytl pro danou topologii přímější kontrolu nad simulací bez nutnosti skokové duplikace a přepínání celých bloků geometrie.

2.6.5 Hybridní přístup k modelování mořské hladiny

Pro zasazení plavidla do realistického prostředí bylo nutné vytvořit mořskou hladinu reagující na pohyb lodi i na padající fragmenty po explozi. Vzhledem k extrémní hardwarové náročnosti objemových simulací kapalin (fluid dynamics) bylo rozhodnuto odložit simulaci celého širého oceánu pomocí částic. Místo toho byl zvolen hybridní přístup kombinující procedurální geometrii a lokální fyzikální výpočet.

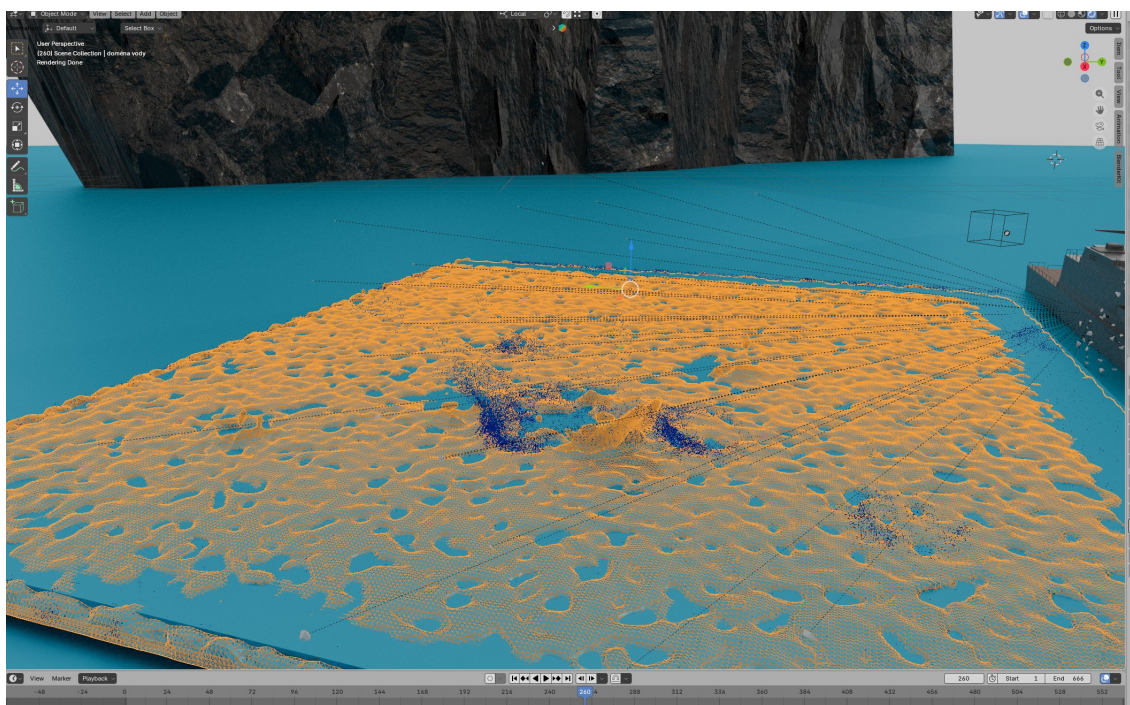
Základní mořská hladina byla vygenerována procedurálně pomocí modifikátoru *Ocean*, který na plošnou geometrii (Plane) aplikuje matematické deformační algoritmy simulující vlnění. Pro vytvoření interakce mezi plujícím trupem lodi a vodou byl využit systém *Dynamic Paint*. V tomto nastavení fungovala plocha oceánu jako plátno (*Canvas*) v režimu *Waves*, zatímco trup plavidla působil jako štětec (*Brush*). Tato metoda umožnila v reálném čase a s minimálními nároky na výpočetní výkon vygenerovat realistickou brázdou a odrazové vlny šířící se od přídě lodi.

2.6.6 Lokální simulace vodní tříště pomocí Mantaflow

Zatímco procedurální oceán dostatečně reprezentoval širé okolí, nedokázal reagovat na drobné dynamické úlomky padající z radarových dómů. Pro vytvoření fotorealistických cákanců a vodní tříště v místě dopadu byl proto implementován fyzikální kapalinový solver *Mantaflow*.

S ohledem na optimalizaci renderovacího času byla vytvořena takzvaná lokální simulační doména (*Local Fluid Domain*). Jednalo se o ohraničený výpočetní prostor definovaný striktně jen pro oblast, kam po explozi dopadaly trosky. Uvnitř této domény byl vytvořen počáteční objem tekutiny (*Flow Object* v režimu *Geometry*), jehož horní plocha exaktně lícovale s výškou vln procedurálního oceánu.

Samotným fragmentům dómů byl přiřazen parametr *Effector* s typem *Collision*, čímž se staly pro kapalinový engine překážkou. Při průniku těchto polygonů s povrchem kapaliny v doméně došlo k fyzikálně přesnému přenosu kinetické energie a vygenerování objemové vodní tříště. Tímto hybridním spojením procedurální nekonečné hladiny a striktně lokální částicové simulace bylo dosaženo vysoké vizuální kvality scény bez neúměrného zatížení paměti počítače a celkového času výpočtu.



Obrázek 22: Simulace vody

Zdroj: vlastní práce

2.7 Destrukce v 3DS MAX

Pro vytvoření realistické destrukce radarových dómů v softwaru 3ds Max bylo nutné primární geometrii rozdělit na desítky nezávislých úlomků, na které bude následně aplikován fyzikální engine. Zatímco Blender disponuje vysoce stabilním integrovaným algoritmem Cell Fracture, nativní řešení v 3ds Max (nástroj *ProCutter* založený na operacích typu Boolean) se při složitějších průnicích ukázalo jako nevhodné. Při pokusech o vícenásobný řez generoval *ProCutter* závažné defekty v topologii sítě, takzvané Boolean artefakty (převrácené normály a chybějící polygony), které znemožňovaly korektní výpočet srážek.

Z tohoto důvodu bylo nutné sáhnout po externím řešení v podobě modulu Fracture Voronoi (MAXScript). Tento skript nahrazuje zastaralý nativní algoritmus pokročilou matematickou metodou Voronoiho tessellace. Výsledkem byla okamžitá a stabilní generace desítek uzavřených polygonálních těles (tzv. watertight meshes) bez jakýchkoliv geometrických chyb, které byly ideálně připraveny pro implementaci fyziky pevných těles. Tento krok jasně demonstruje častou nutnost modulárního rozšiřování 3ds Max pomocí externích skriptů oproti ucelené výbavě softwaru Blender.

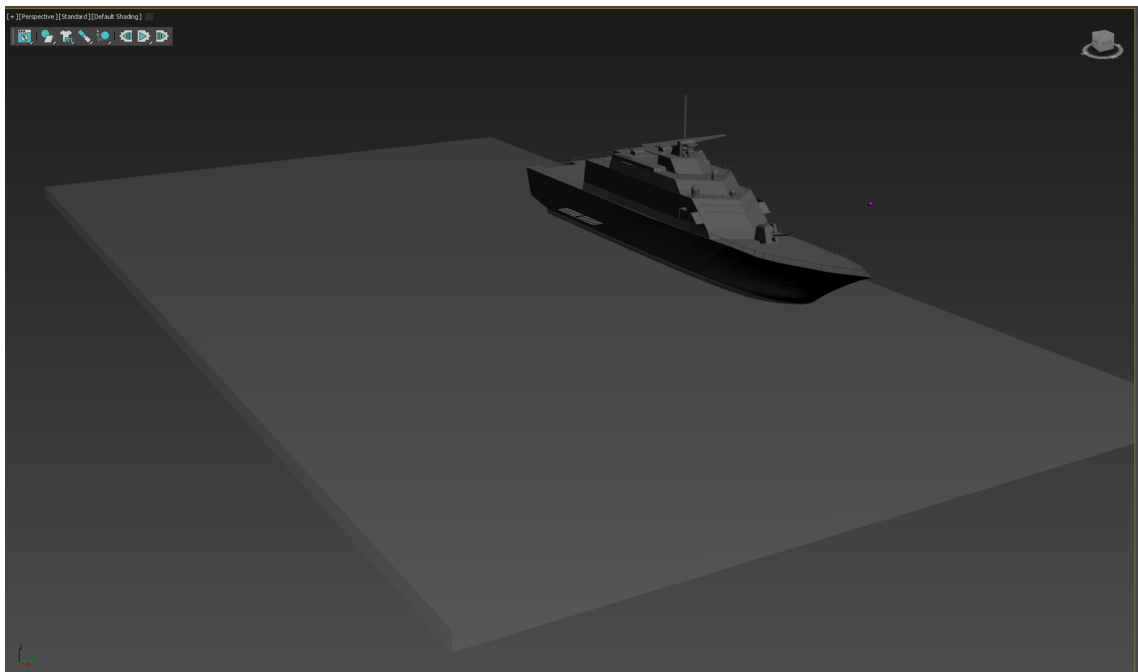
2.8 Simulace vodního prostředí a limity modulu Fluid ve 3DS MAX

Pro účely exaktního srovnání pracovních postupů bylo přistoupeno k replikaci simulace vodní hladiny v prostředí 3ds Max pomocí nativního systému Fluid (využívajícího výpočetní jádro Bifrost). Cílem bylo vytvořit ohraničený statický objem kapaliny pomocí metody Container, do kterého následně dopadnou dynamické fragmenty dómů a vytvoří lokální tříšť.

Během implementace se však plně projevila architektonická nestabilita a specifická omezení tohoto nativního řešení. I přes korektní definici emisního objektu a detailní nastavení parametrů

Voxelové sítě nedokázal solver Fluid inicializovat počáteční stav kapaliny. Detailní analýzou tohoto problému bylo zjištěno, že jádro Bifrost je extrémně citlivé na absolutní měřítko scény (System Unit Scale a Real-World Scale). Pokud bounding box emisního objektu nebo topologie lodi matematicky nekoresponduje s interním nastavením velikosti voxelů (Base Voxel Size), engine simulaci tiše ukončí bez vygenerování jakýchkoliv chybových hlášení (tzv. silent failure).

Dalším znatelným problémem ovlivňujícím plynulost pracovního postupu (workflow) se ukázala být správa dočasné paměti výpočtu (Solver Cache). 3ds Max agresivně ukládá stará simulační data, což vedlo k zobrazování neexistujících částic (tzv. ghosting) přímo v pracovním okně (Viewportu) i po smazání původních emitů. Oproti transparentnímu systému "pečení" (Baking) v softwaru Blender tak 3ds Max vyžaduje neustálé manuální čištění paměti simulace, což znatelně zpomaluje proces ladění (debugging) scény.



Obrázek 23: Příprava na simulaci vody

Zdroj: vlastní práce

2.8.1 Alternativní přístup

Aby bylo možné alespoň částečně otestovat schopnosti enginu vypočítat interakci mezi kapalinou a dynamickými tělesy řízenými modulem MassFX, bylo od zástupné metody Container upuštěno. Místo toho byl emisní objekt přepnut do režimu klasické kontinuální emise (Emitter). Pomocí animace klíčových snímků (Keyframing) byl parametr Enable Liquid Emission po několika úvodních snímcích programově deaktivován, čímž byl simulován jednorázový dopad masy vody na palubu.

Tento alternativní přístup potvrdil, že samotné matematické jádro pro výpočet kolizí funguje v 3ds Max korektně. Vygenerovaná kapalina bezchybně reagovala jak na kinematický trup lodi, tak na padající dynamické úlomky, které částicím úspěšně předaly svou kinetickou energii.

Ačkoliv je tedy kolizní fyzika softwaru na vysoké úrovni, neschopnost programu spolehlivě a intuitivně vygenerovat statickou hladinu bez nutnosti hlubokých zásahů do globálního měřítka

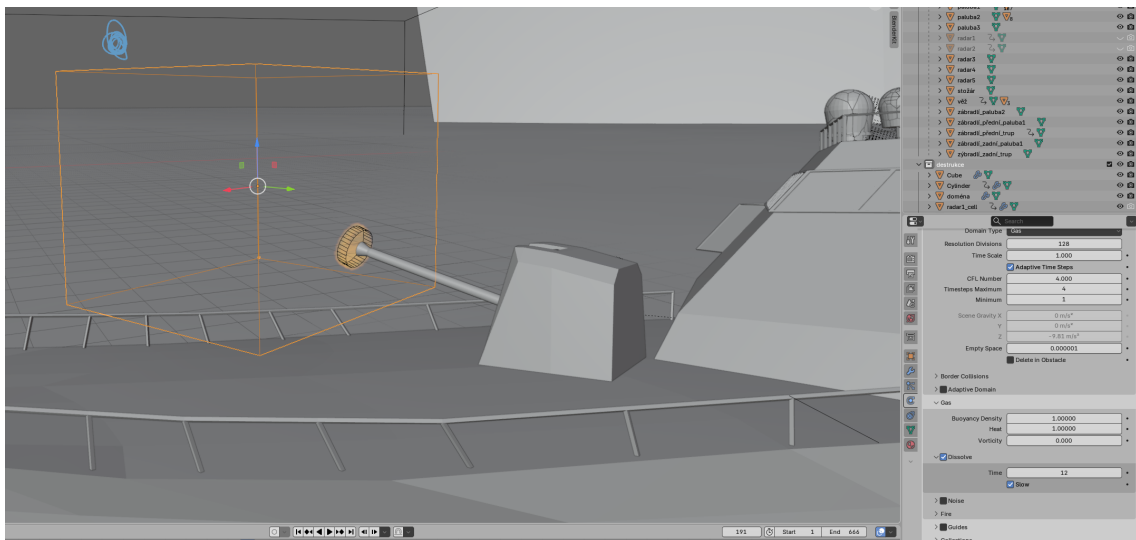
celé topologie odhaluje jeho slabiny při rychlém prototypování (rapid prototyping). Tyto poznatky jasně zdůvodňují, proč se v profesionální produkční praxi 3ds Max pro pokročilé simulace tekutin standardně využívají robustnější externí komerční pluginy (např. Phoenix FD). V přímém srovnání čistých instalací obou programů tak vychází integrovaný modul Mantaflow v Blenderu jako výrazně flexibilnější a uživatelsky přívětivější volba, která lépe odpouští odchylky v topologii a lokálním měřítku.

2.9 Volumetrické simulace a efekty (VFX)

Nedílnou součástí vizualizace bojového nasazení plavidla bylo vytvoření záblesku při výstřelu z hlavně (tzv. muzzle flash). Zde se naplno projevil rozdíl v celkovém pojetí obou testovaných programů z hlediska vizuálních efektů (VFX).

2.9.1 Integrované řešení v softwaru Blender

V prostředí Blenderu byl výstřel realizován prostřednictvím modulu Mantaflow, který kromě kapalin nativně podporuje i plnohodnotné volumetrické simulace plynů a ohně. Pro vytvoření realistického výšlehu plamene byl nastaven objekt s typem emise Fire + Smoke. Klíčovým parametrem byla aplikace vysoké počáteční rychlosti (Initial Velocity), která společně s animací parametru Use Flow v řádu jednotek snímků korektně simulovala okamžitou expanzi plynů z hlavně.



Obrázek 24: Simulace výbuchu

Zdroj: vlastní práce

2.9.2 Absence nativních volumetrických nástrojů v 3DS MAX

Při pokusu o replikaci tohoto jevu v prostředí 3ds Max bylo zjištěno, že software v čisté instalaci nedisponuje žádným moderním volumetrickým solverem pro simulaci plynných jevů. Historické nativní nástroje (jako Atmospheric Fire Effect v kombinaci s Particle Flow) generují pouze neobjemové, plošné iluze (tzv. sprites), které vizuálně ani fyzikálně nevyhovují zadání práce.

Pro vytvoření srovnatelného volumetrického výbuchu v 3ds Max by tak bylo nutné využít buď komerčních pluginů třetích stran (např. Phoenix FD nebo FumeFX), nebo přistoupit k importu předpočítaných VDB dat z externích simulátorů, jako je třeba Blender.

2.10 Finální příprava scény v Blenderu

Před samotným zahájením finálního výpočtu (renderingu) bylo nezbytné celou scénu optimalizovat pro plynulý a bezchybný export. Vzhledem k vysoké výpočetní a časové náročnosti spojené s renderingem volumetrických efektů (oheň) a kapalinových simulací (Mantaflow) byl kladen maximální důraz na minimalizaci nutnosti dodatečných úprav v postprodukcí.

V prostředí softwaru Blender byla pro tyto účely využita nativní funkce automatizovaného stříhu přímo v rámci 3D scény (Bind Camera to Markers). Vzhledem k tomu, že vizualizace vyžadovala prezentaci akce z vícero úhlů, bylo by standardním postupem renderovat několik oddělených obrazových sekvencí a ty následně skládat v externím stříhovém softwaru.

Místo toho byla na globální časovou osu (Timeline) aplikována série řídicích značek (Markers). Ke každé z těchto značek byla dedikována konkrétní kamera ve scéně. Během finálního sekvenčního renderingu tak engine automaticky přepínal aktivní pohledy v přesně definovaných snímcích. Celá animace tak mohla být odeslána k výpočtu jako jedna kontinuální obrazová sekvence PNG souborů, ze kterých rovnou vznikl finální sestřih. Tento postup nejenže eliminoval potřebu základního video editingu v postprodukční fázi, ale především garantoval, že nedojde k renderování nadbytečných snímků.

2.11 Finální příprava scény v 3DS MAX

Obdobně jako v případě softwaru Blender, i scéna v 3ds Max vyžadovala před finálním výpočtem technologickou přípravu a optimalizaci renderovacího enginu Arnold. Vzhledem k přítomnosti komplexních materiálů lomu světla (vodní hladina) a importovaných volumetrických VDB dat (záblesk výstřelu) bylo klíčové nastavení vzorkování (Sampling). Z důvodu zamezení extrémních renderovacích časů musel být nalezen přesný kompromis mezi hodnotami Camera (AA) vzorků a specifickými vzorky pro Transmission (průhlednost) a Volume (objem).

Zásadní rozdíl mezi oběma testovanými programy se projevil v přístupu k automatizaci renderingu více kamer. Zatímco Blender umožňuje nativní přepínání kamerových pohledů přímo na globální časové ose, 3ds Max využívá pro automatizovaný sekvenční výpočet modul Batch Render.

Prostřednictvím tohoto nástroje byla vytvořena výpočetní fronta (Queue), v níž byl každé kameře přiřazen specifický úsek časové osy (Frame Range) a vlastní výstupní cesta pro uložení PNG sekvence. Tento modul zajišťuje bezobslužný průběh renderingu přes noc, nicméně na rozdíl od přístupu v Blenderu neprodukuje jeden kontinuálně sestříhaný výstup. Každý kamerový úhel je vyrenderován jako oddělená obrazová sekvence, což vyžaduje dodatečnou kompletaci v externím postprodukčním softwaru. Z hlediska produkční pipeline tento přístup sice přidává jeden mezikrok (střih), avšak poskytuje vyšší míru kontroly a flexibility v případě selhání výpočtu či nutnosti přerenderovat pouze izolovaný záběr.

2.12 Optimalizace výpočetního času a nastavení Raytracingu

Ačkoliv moderní renderovací enginy Cycles i Arnold využívají technologii Path Tracingu, jejich efektivní nasazení v animaci vyžaduje drastickou optimalizaci hloubky paprsků (Ray Depth/Light Paths) (Pharr et al., 2016). Vzhledem k povaze scény byla největší výpočetní zátěž identifikována v lomu světla skrz vodní hladinu (Transmission) a ve výpočtu rozptylu světla uvnitř volumetrického efektu výstřelu (Volume Bounces).

Aby se předešlo exponenciálnímu nárůstu renderovacího času, byl aplikován optimalizační model: počet odrazů uvnitř volumetrických těles byl snížen na hodnotu 0 až 1 (jelikož samotný efekt výstřelu primárně funguje jako emisivní zdroj), zatímco hodnota pro průchod světla vodou (Transmission) byla zafixována na hodnotě 6, aby nedocházelo k nežádoucímu černání geometrie v místech překrývání vln. Oba programy byly navíc podpořeny adaptivním vzorkováním v kombinaci s postprodukčním algoritmem OpenImageDenoise (OIDN), který na bázi umělé inteligence odstraňuje zbytkový šum bez nutnosti generování extrémního počtu vzorků (Samples).

2.13 Finální rendering a kompletace audiovizuálních výstupů

Závěrečnou fází praktické části práce byl samotný strojový výpočet (rendering) připravených scén a jejich následná kompletace do finálního video formátu. Pro zajištění maximální bezpečnosti dat a zamezení ztráty výpočetního času při případném nestandardním chování hardwaru byly oba programy nastaveny na export do bezztrátových obrazových sekvencí (formát PNG), což odpovídá zavedeným průmyslovým standardům. Až v této finální fázi se naplno projevil propastný rozdíl v celkové filozofii obou softwarů.

2.13.1 Integrovaná postprodukce v softwaru Blender

V prostředí softwaru Blender byl rendering realizován prostřednictvím raytracingového enginu Cycles za využití GPU akcelerace. Zásadní technologickou výhodou se v tomto kroku ukázala přítomnost integrovaného modulu Video Sequence Editor (VSE). Jakmile byla obrazová sekvence vyrenderována, bylo možné ji okamžitě načíst do nativní stříhové osy přímo uvnitř Blenderu. Bez nutnosti spouštět jakýkoliv externí program tak mohla být sekvence zkontrolována, časově upravena a rovnou enkódována přes knihovnu FFmpeg do finálního kontejneru MPEG-4 (s využitím kodeku H.264). Tento All-in-One přístup drasticky zefektivnil poslední fázi projektu a potvrdil pozici Blenderu jako vysoce autonomního nástroje.

2.13.2 Nutnost externí pipeline u 3DS MAX

Oproti tomu práce v 3ds Max s enginem Arnold demonstrovala klasický studiový přístup založený na fragmentované produkční pipeline. Přestože optimalizované nastavení vzorkování udrželo výpočetní časy na přijatelné úrovni, po dokončení modulu Batch Render vznikla pouze surová data (složky s obrazovými sekvencemi jednotlivých kamer). Jelikož 3ds Max nedisponuje plnohodnotným integrovaným modulem pro nelineární stříh a enkódování videa, vyvstala technologická nutnost použít externí postprodukční software, v tomto případě byl použit Blender.

Závěr

Cílem této bakalářské práce bylo provést objektivní komparaci dvou 3D softwarů, open-source programu Blender a komerčního standardu Autodesk 3DS MAX, na případové studii tvorby vizuálních efektů (VFX) bojového nasazení plavidla. Práce nehodnotila primárně finální vizuální kvalitu, ale zaměřila se především na technologické rozdíly, uživatelskou přívětivost, hardwarovou náročnost a celkovou logiku produkční pipeline.

V počátečních fázích produkce, zahrnujících polygonální modelování a definici povrchových vlastností, oba programy prokázaly plnou podporu moderních průmyslových standardů. Využití PBR (Physically Based Rendering) materiálů zajistilo technicky identické chování světla v obou aplikacích, avšak je nutno zmínit, že opravdový profesionál z praxe by používal profi režim, který zásadně zjednodušuje práci v 3DS MAX. Přesto Blender v této fázi nabídl výrazně plynulejší iterační proces díky integraci real-time renderovacího enginu Eevee, který poskytuje okamžitou vizuální zpětnou vazbu bez nutnosti zdlouhavého předpočítávání, na které spoléhá nativní zobrazení ActiveShade v 3ds Max.

Zásadní technologické rozdíly a limity byly identifikovány v oblasti fyzikálních a volumetrických simulací. Blender se prezentoval jako vysoce autonomní All-in-One balíček. Jeho nativní modul Mantaflow umožnil v rámci jednoho uživatelského rozhraní realizovat jak interakci rigidních těles s kapalinou, tak i vysokorychlostní volumetrickou emisi plynů (záblesk výstřelu z hlavně). Naproti tomu architektura 3DS MAX v čisté instalaci nedisponuje moderními volumetrickými solvery. Řešení tohoto deficitu by si vyžádalo využití pokročilé filmové pipeline spočívající v importu předpočítaných VDB dat z externího simulátoru, například Blenderu a jejich následném renderování prostřednictvím modulu Arnold Volume.

Tato skutečnost se promítla i do závěrečné fáze kompletace výstupů. Zatímco Blender umožnil automatizovaný střih kamer a následný export přímo do finálního video kontejneru prostřednictvím integrovaného Video Sequence Editoru, 3DS MAX si i po optimalizaci sekvenčního výpočtu (Batch Render) vyžádal kompletaci v externím postprodukčním softwaru.

Na základě provedené případové studie lze vyvodit jednoznačný závěr: ani jeden ze softwarů nelze plošně označit za lepší, neboť každý slouží odlišné produkční filozofii. Blender vyniká jako komplexní, nezávislý ekosystém, který je díky své rychlosti, real-time nástrojům a integrovaným simulátorům ideální volbou pro nezávislé tvůrce, malá studia a rapid prototyping. Autodesk 3DS MAX naopak reflektuje logiku velkých filmových a herních studií. Nesnaží se poskytovat veškeré simulační nástroje nativně, ale funguje spíše jako extrémně robustní centrální uzel a finální montážní linka, která primárně spoléhá na ekosystém vysoce specializovaných komerčních pluginů a na import dat z dedikovaných softwarů.

Práce tak úspěšně prokázala, že pro efektivní tvorbu VFX nezáleží pouze na znalosti konkrétních funkcí, ale především na schopnosti zvolit správnou softwarovou architekturu pro daný typ a rozsah projektu. Je však nezbytné podotknout, že dříve striktní hranice mezi „nástrojem pro nezávislé tvůrce“ a „průmyslovým standardem“ se v současnosti postupně tenčí. Důkazem je rostoucí trend integrace Blenderu do produkčních pipeline velkých filmových společností, jakým je například studio Framestore při tvorbě vizuálních efektů pro hranou adaptaci snímku Jak vycvičit draka. Ačkoliv v těchto případech Blender neslouží jako výhradní softwarové řešení pro celý projekt, je cíleně nasazován na specifické a vysoce komplexní úkony – jako je například vývoj a modelování stěžejních CGI draků. (Seymour, 2023) Tato skutečnost potvrzuje, že efektivní VFX produkce budoucnosti nestojí na dominanci jediného programu, ale na promyšleném propojování těch nejlepších nástrojů, které současný trh nabízí.

Tento postupný posun v preferencích velkých studií úzce souvisí s diametrálně odlišnými modely vývoje obou platforem. Blender v současnosti těží z mimořádně agilního open-source přístupu, který je masivně finančně i technologicky podporován předními lídry globálního trhu prostřednictvím Blender Development Fund. Výsledkem je vysoce dynamický vývojový cyklus, který dokáže téměř okamžitě reagovat na moderní produkční trendy a inovace. Naproti tomu licenční a komerční model softwaru Autodesk 3ds Max začíná narážet na technologické limity své starší systémové architektury. Přestože se jedná o prémiový placený standard, tempo jeho nativních inovací je v profesionální sféře často vnímáno jako stagnující ve srovnání s razantním vývojem jeho open-source konkurence. Tento kontrastní vývojový trend je jedním z hlavních faktorů, proč i tradiční a prestižní VFX studia postupně modifikují své produkční pipeline a integrují progresivněji se vyvíjející nástroje.

Seznam použité literatury

Blender.org. (n.d.). *History*. [online] Dostupné z: <https://www.blender.org/about/history/> [cit. 2024-20-12].

GarageFarm.NET. (n.d.). A brief history of 3ds Max. [online] Dostupné z: <https://garagefarm.net/blog/3ds-max-versatility-to-the-max> [cit. 2024-20-12].

Blender.org. (n.d.). *Reference Manual*. [online] Dostupné z: <https://docs.blender.org/manual/en/latest/index.html> [cit. 2025-7-1].

CHOPINE, Ami. (2011). *3D art essentials: the fundamentals of 3D modeling, texturing, and animation*. Taylor & Francis. ISBN 978-0-240-81170-4.

VAUGHAN, William. (2012). *Digital Modeling*. New Riders. ISBN 978-0-321-70089-6.

MCDERMOTT, Wes. (2018). *The PBR Guide: A Handbook for Physically Based Rendering*. [online] Dostupné z: <https://substance3d.adobe.com/tutorials/courses/the-pbr-guide-part-1> [cit. 2026-04-14].

BRIDSON, Robert. (2015). *Fluid simulation for computer graphics*. AK Peters/CRC Press. ISBN 978-1-4822-3283-7.

AUTODESK. (2024). *3ds Max 2024 Help*. [online] Dostupné z: <https://help.autodesk.com/view/3DSMAX/2024/ENU/> [cit. 2026-04-14].

PHARR, Matt, Jakob WENZEL a Humphreys GREG. (2016). *Physically based rendering: From theory to implementation*. Morgan Kaufmann. ISBN 978-0-12-800645-0.

SEYMOUR, Mike. (2023). *Framestore and the adoption of Blender*. [online] fxguide. Dostupné z: <https://www.fxguide.com/ffeatured/framestore-and-the-adoption-of-blender/> [cit. 2026-04-14].