

**Česká zemědělská univerzita v Praze**

**Provozně ekonomická fakulta**

**Katedra informačního inženýrství**



## **Diplomová práce**

**Srovnání herních enginů Unreal Engine a Unity**

**Bc. Martin Khol**

**© 2024 ČZU v Praze**

## ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Martin Khol

Informatika

Název práce

**Srovnání herních engineů Unreal Engine a Unity**

Název anglicky

**Comparison of game engines Unreal Engine and Unity**

### Cíle práce

Teoretická práce bude vytvořena na základě studia literárních zdrojů zaměřených na Unreal Engine a Unity. Bude provedena analýza jejich architektury a způsobu použití.

Dílním cílem práce bude vyhodnotit, jak tyto herní enginey podporují různé typy her a herních žánrů. Budou zkoumány možnosti tvorby 2D a 3D her, virtual reality (VR) a rozšířené reality (AR). Bude zkoumáno také, jakým způsobem enginey umožňují implementaci multiplayerových funkcí a podporu různých platform, jako jsou PC, konzole a mobilní zařízení.

Dalším důležitým aspektem práce bude porovnání dostupných nástrojů pro tvorbu obsahu a jejich uživatelskou přívětivost. Bude se zkoumat, jak snadno lze vytvářet a editovat herní prvky, animace, modely, zvuky a efekty v obou enginech. Bude také provedeno srovnání programovacích jazyků a skriptovacích nástrojů, které jsou používány při vývoji her v Unreal Engine a Unity.

V praktické části budou vytvořeny a provedeny scénáře za účelem porovnání funkcí a vlastností obou engineů. Tyto scénáře budou vypracovány jako prototypy her na kterých budou vyzkoušeny vlastnosti herních engineů.

### Metodika

Bude provedeno studium literatury, odborných článků, dokumentace a online zdrojů týkajících se Unreal Engine a Unity. Toto studium umožní získat teoretický základ pro porozumění architektury a funkcionalit obou herních engineů. Na základě získaných znalostí bude provedena podrobná analýza architektury a způsobu použití Unreal Engine a Unity. Tato analýza poskytne hlubší porozumění interním strukturám obou engineů a jejich přístupu k vývoji her.

Následně bude provedeno porovnání možností obou engineů při podpoře různých typů her a herních žánrů. Budou zkoumány možnosti vytváření a editace herních prvků, animací, modelů, zvuků a efektů v obou enginech. Také budou porovnány programovací jazyky a skriptovací nástroje používané při vývoji her v Unreal Engine a Unity.

V praktické části práce budou vytvořeny prototypy her, ve kterých budou testovány a porovnávány různé aspekty Unreal Engine a Unity.

Na závěr budou diskutovány jejich přednosti a nevýhody, a na základě toho budou formulována doporučení pro vývojáře her při výběru vhodného herního engineu pro jejich projekty.

## Doporučený rozsah práce

60–80 stran

## Klíčová slova

srovnání enginů, Unity, Unreal Engine, vývoj hry, prototyp hry

---

## Doporučené zdroje informací

HOCKING, Joseph. Unity in Action: Multiplatform Game Development in C#. 3rd Edition. New York: Manning Publications, 2022. ISBN 1617299332.

MOHD, Tauheed Khan, Fernando BRAVO-GARCIA, Landen LOVE, Mansi GUJADHUR a Jason NYADU. Analyzing Strengths and Weaknesses of Modern Game Engines. International Journal of Computer Theory and Engineering. 2023, 15(1), 54–60. ISSN 17938201. Dostupné z: doi:10.7763/IJCTE.2023.V15.1330

SALAMA, Ramiz a Mohamed ELSAYED. A live comparison between Unity and Unreal game engines. Global Journal of Information Technology Emerging Technologies. 2021, 1(11), 01–07. Dostupné z: doi:10.18844/gjit.v11i1.5288

SEWELL, Brenden. Blueprints Visual Scripting for Unreal Engine: Build professional 3D games with Unreal Engine 4's Visual Scripting system. Birmingham: Packt Publishing, 2015. ISBN 1785286013.

UNITY TECHNOLOGIES. Unity User Manual [online]. San Francisco: Unity Technologies, 2022. Dostupné z: <https://docs.unity3d.com/>

Unreal Engine Documentation [online]. Cary, NC: Epic Games, 2023. Dostupné z: <https://docs.unrealengine.com/en-US/index.html>

---

## Předběžný termín obhajoby

2023/24 LS – PEF

## Vedoucí práce

Ing. Dana Vynikarová, Ph.D.

## Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 28. 11. 2023

**Ing. Martin Pelikán, Ph.D.**

Vedoucí katedry

Elektronicky schváleno dne 9. 2. 2024

**doc. Ing. Tomáš Šubrt, Ph.D.**

Děkan

V Praze dne 25. 03. 2024

### **Čestné prohlášení**

Prohlašuji, že svou diplomovou práci „Srovnání herních enginů Unreal Engine a Unity“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 31.3.2024

---

### **Poděkování**

Rád bych touto cestou poděkoval Ing. Daně Vynikarové, Ph.D. za odborné vedení této práce.

# Srovnání herních enginů Unreal Engine a Unity

## Abstrakt

Tato diplomová práce se zabývá srovnáním vlastností herních enginů v rámci herního vývoje. Práce je rozdělena na dvě části.

První část popisuje teoretická východiska práce. Zabývá se problematikou herního vývoje a rolí herních enginů. Jsou detailně představeny herní enginy Unity a Unreal Engine a jejich nástroje. V závěru teoretické části je představena role prototypů v herním vývoji.

V druhé části práce jsou představeny vytvořené prototypy herních aplikací a provedeny srovnávací testy obou herních enginů.

Výstupem práce jsou doporučení a poučení pro výběr vhodného herního enginu při vývoji herní aplikace. Zjištěno bylo, že aplikace vytvořeny v Unreal Engine nabízí vyšší vizuální kvalitu za cenu větší náročnosti na hardwarové zdroje a aplikace vytvořeny v Unity umožňuje lépe podporovat tvorbu dvouprostorových her a je vhodnější pro platformy s méně výkonným hardwarem.

**Klíčová slova:** srovnání enginů, Unity, Unreal Engine, vývoj hry, prototyp hry

# Comparison of game engines Unreal Engine and Unity

## Abstract

The thesis deals with the comparison of the properties of game engines in the context of game development. The thesis is divided into two parts.

The first part describes the theoretical background of the thesis. It deals with the issues of game development and the role of game engines. Game engines Unity and Unreal Engine and their tools are introduced in detail. At the end of the theoretical part the role of prototypes in game development is presented.

In the second part of the thesis, prototypes of game applications are presented, and comparative tests of both game engines are performed.

The output of the thesis are recommendations and lessons learned for selecting an appropriate game engine when developing a game application. It was found that applications created in the Unreal Engine offer higher visual quality at the cost of greater demand on hardware resources, and applications created in Unity allows better support for the creation of dual-space games and is more suitable for platforms with less powerful hardware.

**Keywords:** game engine comparison, Unity, Unreal Engine, game development, game prototype

# Obsah

<b>1 Úvod.....</b>	<b>11</b>
<b>2 Cíl práce a metodika .....</b>	<b>13</b>
2.1 Cíl práce .....	13
2.2 Metodika .....	13
<b>3 Teoretická východiska .....</b>	<b>15</b>
3.1 Herní vývoj.....	15
3.1.1 Tvorba her.....	15
3.1.2 Proces vývoje.....	15
3.1.3 Herní engine.....	17
3.2 Komponenty herního enginu.....	18
3.2.1 Renderování .....	18
3.2.2 Animace .....	20
3.2.3 Fyzikální engine.....	22
3.2.4 Umělá inteligence .....	22
3.2.5 Scriptování .....	23
3.2.6 Zvuk.....	24
3.3 Unity.....	25
3.3.1 Historie.....	25
3.3.2 Cena .....	26
3.3.3 Budoucnost .....	27
3.3.4 Dostupnost návodů a assetů.....	27
3.3.5 Příklady her.....	28
3.3.6 Kompatibilita s různými platformami.....	28
3.3.7 Programovací jazyky .....	28
3.4 Unreal Engine.....	29
3.4.1 Historie.....	29
3.4.2 Cena .....	30
3.4.3 Budoucnost .....	31
3.4.4 Dostupnost návodů a assetů.....	31
3.4.5 Příklady her.....	32
3.4.6 Kompatibilita s platformami.....	33
3.4.7 Programovací jazyky .....	33
3.5 Tvorba Prototypů .....	35
3.5.1 Co je to prototyp .....	35
3.5.2 Tvorba prototypu v Unity .....	36



3.5.3	Tvorba prototypu v Unreal Engine .....	37
3.5.4	Porovnání tvorby prototypů .....	38
3.6	Implementace online her pro více hráčů .....	39
3.6.1	Multiplayerový vývoj a nástroje v Unity .....	39
3.6.2	Multiplayerový vývoj v Unreal Engine .....	41
3.6.3	Shrnutí online vývoje v obou platformách .....	42
<b>4</b>	<b>Vlastní práce .....</b>	<b>44</b>
4.1	Verze enginů .....	44
4.2	Výkon fyzikálních výpočtů .....	44
4.2.1	Metodika testování .....	45
4.2.2	Postup při průběhu testu .....	45
4.2.3	Unity .....	45
4.2.4	Unreal Engine .....	48
4.3	Výkon grafického renderování .....	49
4.3.1	Metodika testování .....	50
4.3.2	Postup při průběhu testu .....	50
4.3.3	Výsledky testu vykreslování statických modelů .....	51
4.3.3.1	Unity .....	51
4.3.3.2	Unreal Engine .....	52
4.3.4	Výsledky testu vykreslování dynamických modelů .....	53
4.3.4.1	Unity .....	53
4.3.4.2	Unreal Engine .....	54
<b>5</b>	<b>Výsledky a diskuse .....</b>	<b>56</b>
5.1	Výsledky porovnání výkonu herních enginů Unreal Engine a Unity .....	56
5.2	Limity výsledků .....	60
5.3	Interpretace výsledků .....	60
5.4	Finanční srovnání .....	61
5.5	Shrnutí Unity .....	62
5.6	Shrnutí Unreal Engine .....	62
5.7	Doporučení .....	62
5.8	Přínos .....	63
<b>6</b>	<b>Závěr .....</b>	<b>65</b>
<b>7</b>	<b>Seznam použitých zdrojů .....</b>	<b>66</b>
<b>8</b>	<b>Seznam obrázků, tabulek, grafů a zkratk .....</b>	<b>71</b>
8.1	Seznam obrázků .....	71
8.2	Seznam tabulek .....	71

8.3	Seznam grafů.....	71
-----	-------------------	----

# 1 Úvod

V dnešním světě jsou počítačové hry součástí našich každodenních životů. Setkáváme se s nimi v našich osobních životech i na veřejnosti. Každý z nás vlastní mobilní zařízení, které jsou schopna simulovat komplexní světy do kterých se můžeme ponořit cestou do práce nebo za ošklivého počasí.

Tato práce se zabývá jedním aspektem procesu vývoje těchto aplikací. Různých nástrojů a programů využívaných v herním vývoji je mnoho, ale nejdůležitější z nich představují herní enginy. Jde o komplexní programy, které musí být univerzální natolik, aby umožnili sestavit mnoho různých žánrů her a při tom zároveň musí být srozumitelné pro uživatele s různými technickými znalostmi a přístupem.

Tato práce představuje herní enginy a porovnává je. Vybrány byly dva největší herní enginy z pohledu podílu na trhu s herními aplikacemi. Těmito enginy jsou Unity od stejnojmenné společnosti a Unreal Engine od společnosti Epic Games. Popisuje vlastnosti jednotlivých enginů. Konkrétně se zabývá programovacími jazyky, finančním modelem a dostupnými nástroji, které jsou součástí herních enginů. Dále jsou představeny možnosti vývoje online her pro více hráčů a podpora vývoje pro různé platformy.

Pro konkrétní demonstraci určitých vlastností herních enginů jsou sestaveny prototypy určené k jejich měření. Tímto jsou demonstrovány schopnosti simulace fyzikálních interakcí a vykreslování detailních statických a dynamických modelů.

Z pohledu herního vývojáře představuje volba herního enginu jedno z nejdůležitějších rozhodnutí. Dopad má toto rozhodnutí na celý proces vývoje i na výsledný produkt. V dnešní době nabízí velké herní enginy spousty podobných funkcionalit, ale také jsou mezi nimi výrazné rozdíly. Tato práce si dává za cíl představit odlišnosti těchto enginů a usnadnit volbu při jejich použití.

V první části práce je vysvětlen proces herního vývoje a role, kterou v tomto procesu představují herní enginy. Následně jsou herní enginy představeny a vysvětlen výběr Unity a Unreal Engine ke srovnání v této práci. Další dvě kapitoly představují Unity a Unreal Engine. Rozebrány jsou do detailu důležité aspekty herních enginů. Závěr první části se věnuje prototypům a jejich roli v herním vývoji.

V druhé části práce jsou představeny testovací prototypy demonstrující vlastnosti herních enginů Unity a Unreal Engine. Tyto aplikace jsou použity k testování výkonu těchto enginů ve fyzikálním testu a testech vykreslování dynamických a statických objektů.

Závěrem jsou diskutovány poznatky z předchozích kapitol a shrnuty do konkrétních doporučení.

Přirozeným prostředím pro tyto herní enginy a jejich uživatele je internet, a proto velká část děl využívaných pro tuto práci je k dispozici online. Zároveň jsou ale využity mnohé z dostupných knih a článků o herním vývoji.

## **2 Cíl práce a metodika**

### **2.1 Cíl práce**

Teoretická práce bude vytvořena na základě studia literárních zdrojů zaměřených na Unreal Engine a Unity. Bude provedena analýza jejich architektury a způsobu použití.

Dílčím cílem práce bude vyhodnotit, jak tyto herní enginy podporují různé typy her a herních žánrů. Budou zkoumány možnosti tvorby 2D a 3D her, virtuální reality (VR) a rozšířené reality (AR). Bude zkoumáno také, jakým způsobem enginy umožňují implementaci multiplayerových funkcí a podporu různých platforem, jako jsou PC, konzole a mobilní zařízení.

Dalším důležitým aspektem práce bude porovnání dostupných nástrojů pro tvorbu obsahu a jejich uživatelskou přívětivost. Bude se zkoumat, jak snadno lze vytvářet a editovat herní prvky, animace, modely, zvuky a efekty v obou enginu. Bude také provedeno srovnání programovacích jazyků a skriptovacích nástrojů, které jsou používány při vývoji her v Unreal Engine a Unity.

V praktické části budou vytvořeny a provedeny scénáře za účelem porovnání funkcí a vlastností obou enginů. Tyto scénáře budou vypracovány jako prototypy her, na kterých budou vyzkoušeny vlastnosti herních enginů.

### **2.2 Metodika**

Bude provedeno studium literatury, odborných článků, dokumentace a online zdrojů týkajících se Unreal Engine a Unity. Toto studium umožní získat teoretický základ pro porozumění architektury a funkcionalit obou herních enginů. Na základě získaných znalostí bude provedena podrobná analýza architektury a způsobu použití Unreal Engine a Unity. Tato analýza poskytne hlubší porozumění strukturám obou enginů a jejich přístupu k vývoji her.

Následně bude provedeno porovnání možností obou enginů při podpoře různých typů her a herních žánrů. Budou zkoumány možnosti vytváření a editace herních prvků, animací, modelů, zvuků a efektů v obou enginů. Také budou porovnány programovací jazyky a skriptovací nástroje používané při vývoji her v Unreal Engine a Unity.

V praktické části práce budou vytvořeny prototypy her, ve kterých budou testovány a porovnávány různé aspekty Unreal Engine a Unity.

Na závěr budou diskutovány jejich přednosti a nevýhody, a na základě toho budou formulována doporučení pro vývojáře her při výběru vhodného herního enginu pro jejich projekty.

## 3 Teoretická východiska

### 3.1 Herní vývoj

#### 3.1.1 Tvorba her

Herní vývoj je proces navrhování, vytváření a vydávání her pro různé platformy (Pulse College, 2023). U největších titulů z pohledu rozpočtu a velikosti studia může trvat mnoho let, na příklad 8 let u titulu „Cyperpunk 2077“ od společnosti CD Projekt Red (Cook, 2023).

Herní studia mají desítky až stovky pracovníků z různých odvětví. Příkladem obvyklých profesí v herním vývoji jsou výtvarníci konceptů, scénáristé, herní designéři, programátoři enginu a programátoři her. V každém herním studiu jsou 3D umělci, jako jsou modeláři, designéři postav, návrháři herních systémů, vývojáři herních aplikací a další. Návrháři materiálů, riggeři, lidé, kteří se starají o nastavení osvětlení, částicových efektů, simulace, postprocesních efektů, pak jsou tu animátoři a režiséři, kteří celý proces řídí. (Dvořáková, 2023)

Tvorba větších titulů je v mnohém srovnatelná s natáčením filmu. Je to velmi nákladný a náročný proces, který zaměstnává mnoho různých profesí, zabere spoustu času. Další podobností s filmovým odvětvím je, že přes vydané náklady je často úspěch herních projektů nejistý. Po vydání se hra může hráčům líbit, ale také nemusí.

#### 3.1.2 Proces vývoje

Na začátku vývoje je nápad. Něco inovativního, co hráče přitáhne. Může to být zcela nový přístup nebo starý osvědčený koncept s několika novými inovativními prvky. Každá hra potřebuje příběh, postavy, které příběh prožívají, a prostředí, ve kterém se odehrává. Autoři se musí rozhodnout o vizuálním stylu hry. Bude vypadat realisticky nebo kresleně? Pro zodpovězení takovýchto otázek je důležité specifikovat cílovou skupinu pro kterou je hra určena.

Podle (BASLER, 2016) mohou být hry děleny na 12 základních žánrů:

**Akční hry** – Jedná se především o 3D hry, kde je základním principem zneškodnit protivníka.

**RPG hry** – Neboli role-playing games. Českým ekvivalentem je hra na hrdiny. Základním prvkem tohoto žánru je hrdina (postava určité hry), který se pohybuje v herním

virtuálním světě. Za procházení a postup světem se postupně vylepšuje a získává nové schopnosti a bonusové body

**Strategické hry** – Základním požadavkem těchto her je strategické myšlení. Hráč disponuje určitým městem, základnou či územím, o které se musí ekonomicky starat a rozvíjet je. Často jsou na hrací mapě protivníci, vyhraje hráč, který má nejlepší strategii. Hrací plocha se nám zobrazuje z ptáčích perspektivy.

**Adventure** – Základním prvkem hry detektivní příběh, kde musí hráč řešit různé hádanky, rébusy a plnit úkoly, aby se mohl v příběhu posunout dál.

**Sportovní hry** – Jedná se o simulace sportů. Často jsou hry konstruovány zjednodušeně, aby bylo možné pohodlné ovládání na úkor přesné simulace sportu.

**Simulace** – Cílem simulátoru je co nejpřesnější napodobení určité činnosti nebo jevu.

**Závodní hry** – Základním principem těchto her je dostat se do cíle první. Ovládáme vozidlo obvykle pomocí klávesnice, či nějakého speciálního periferního zařízení.

**Taneční/hudební hry** – Jedná se o hry, které ovládáme pohybem či používáním speciálních ovladačů.

**Bojové hry** – Základním principem je porazit protivníka v určitém bojovém umění. Obvykle se dva hráči pohybují v areně a různými kombinacemi útoků se snaží zneškodnit protihráče, ať už lidského, nebo počítačem řízeného.

**Logické hry** – Jak již název vypovídá, jsou to hry, které vyžadují logické myšlení hráče, skládají se z různých hádanek. Logické hry jsou obvykle jednoduše graficky zpracované. Hra se často skládá z úrovní, kterými hráč postupně prochází, přičemž se obtížnost pro dokončení úrovně zvyšuje.

**Vzdělávací hry** – Jedná se o nejméně populární počítačové hry, kdy je základním prvkem těchto her edukace hráče.

Když je jasný žánr, stylizace a náplň hry. Proces pokračuje prototypováním hrátelnosti a tvorbou assetů, které prostředí vyplní. Konceptní umělci obvykle kreslí obrázky na základě nápadů výtvarných ředitelů a modeláři pak vytvářejí postavy nebo prostředí na základě těchto konceptních obrázků. Designéři úrovní tyto 3D modely převezmou a vytvoří s nimi hrátelné herní prostředí. Vytvářejí skripty, aby byla úroveň interaktivní, a přidávají postavy s umělou inteligencí. (Jeannie, 2012)



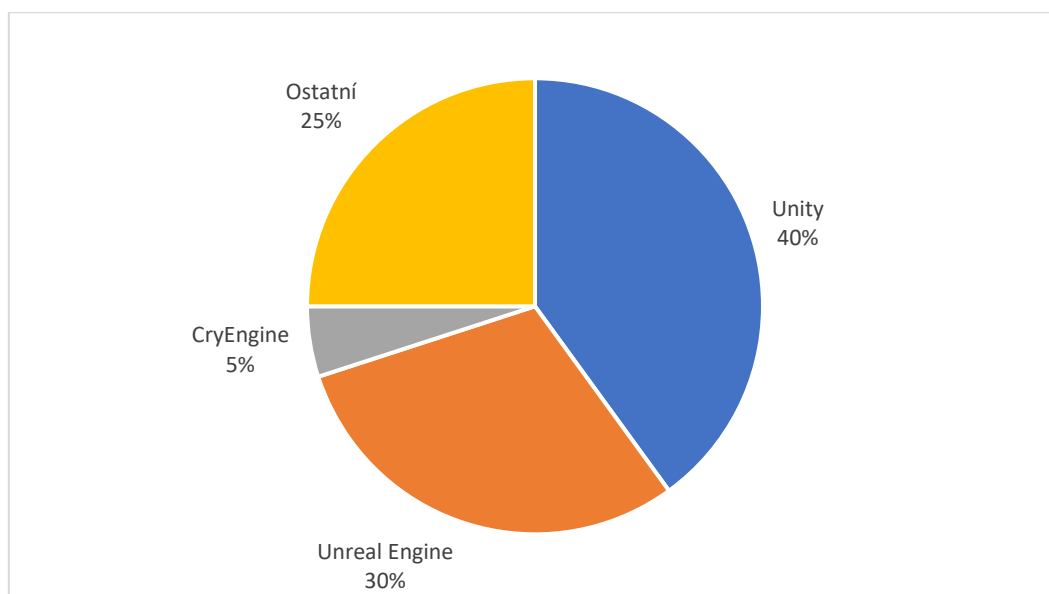
### 3.1.3 Herní engine

Až dosud nebyla zmíněna klíčová složka herního vývoje. Pro úspěšnou spolupráci je třeba, aby existovalo jednotné společné prostředí pro programátory a výtvarníky. Musí existovat komponenta, která vykresluje obraz, stará se o herní logiku, umělou inteligenci, zvuk a multiplayer online komunikaci. Tato základní komponenta se nazývá herní engine.

Od herních enginů jsou vysoká očekávání. Zaprvé musí vykreslovat objekty v reálném čase. Komunikují s hardwarem počítače a vytváří prostředí pro umělce a designéry k zobrazení a práci s jejich assety bez nutnosti znalosti programovacích jazyků. Modely mají různé materiály, textury, normálové mapy. Ve scéně jsou také světla, vizuální efekty a nejdůležitěji kamera, kterou hráč vidí scénu. (Andrýsek, 2019)

Dále musí herní engine spravovat animace. Existuje více druhů animací: jednoduché měnící parametry v čase (pozice, rotace, barva), skeletové animace (pohybují postavy pomocí kostí).

Existuje mnoho herních enginů. Například Unity, Unreal Engine, CryEngine, Amazon Lumberyard, Godot, Creation Engine a další. Největší z pohledu podílu na trhu jsou Unity a Unreal Engine. Obrázek 1 obsahuje základní přehled o podílech na trhu.

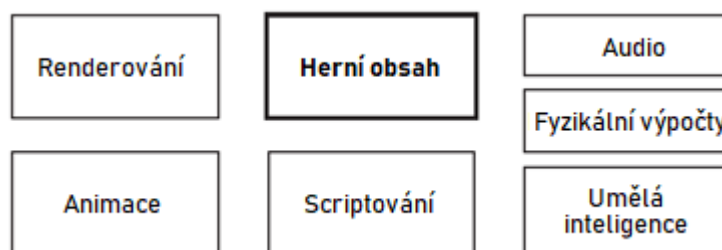


Obrázek 1 Rozložení trhu herních enginů (Sawant, 2023)

## 3.2 Komponenty herního enginu

Herní enginy se skládají ze sady nástrojů a samotné runtime komponenty. Nástroje slouží používají vývojáři k sestavení úrovní, vytváření animací, nastavení zvuků a další. Runtime komponenta pak představuje komplexní program, na kterém vytvořené hry „běží“. Herní enginy používají vrstvenou architekturu, kde vyšší vrstvy jsou závislé na nižších. Nižší vrstvy jsou blíže k hardware a starají se například o práci s pamětí a soubory. Vyšší vrstvy mají na starosti vyšší úroveň zpracování, jako vizuální efekty, animace, kamera a specifická herní logika. (Gregory, 2019)

V následujících podkapitolách jsou zběžně představeny základní systémy z pohledu uživatele herního enginu. Obrázek 2 představuje tyto systémy. (H. Eberly, 2005).



Obrázek 2 Základní komponenty herního enginu (H. Eberly, 2005)

### 3.2.1 Renderování

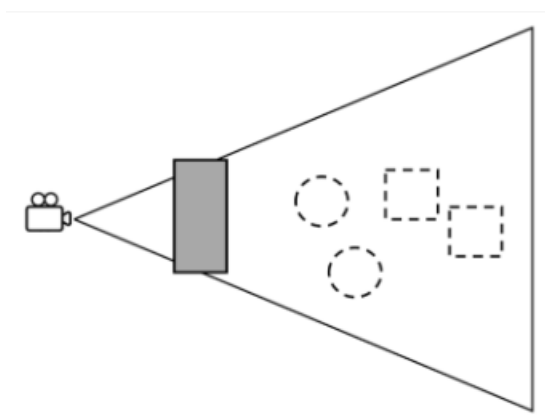
Při zobrazování 3D objektů na obrazovce, používá herní aplikace grafické API, typicky se jedná o OpenGL nebo DirectX. Tato rozhraní umožňují ovládnutí grafické karty a zobrazení požadovaných objektů. Tyto API jsou na nízké úrovni. Pro snadný vývoj her s nimi není žádáno pracovat napřímo, kvůli komplexitě programování na nižší úrovni.

Renderovací engine poskytuje funkce renderování doslova stisknutím tlačítka. Stačí nastavit objekty ve scéně, nastavit kameru, osvětlení, materiály a textury. Když je hra kompilována, renderovací engine se postará o správné zobrazení naší scény. Obvykle je dodáván s řadou shaderů, které mohou být použity k simulaci různých materiálů nebo efektů.

Renderovací engine je nejdůležitější částí herního enginu. Vedle herního mechanismu je hodnocena vizuální prezentace hry. Renderování zpravidla zabírá mnoho výpočetního času. Není neobvyklé, že více než 90 % celkového výpočtu je právě renderování. Zároveň hra musí fungovat dostatečně rychle i na zařízeních s nižším výkonem,

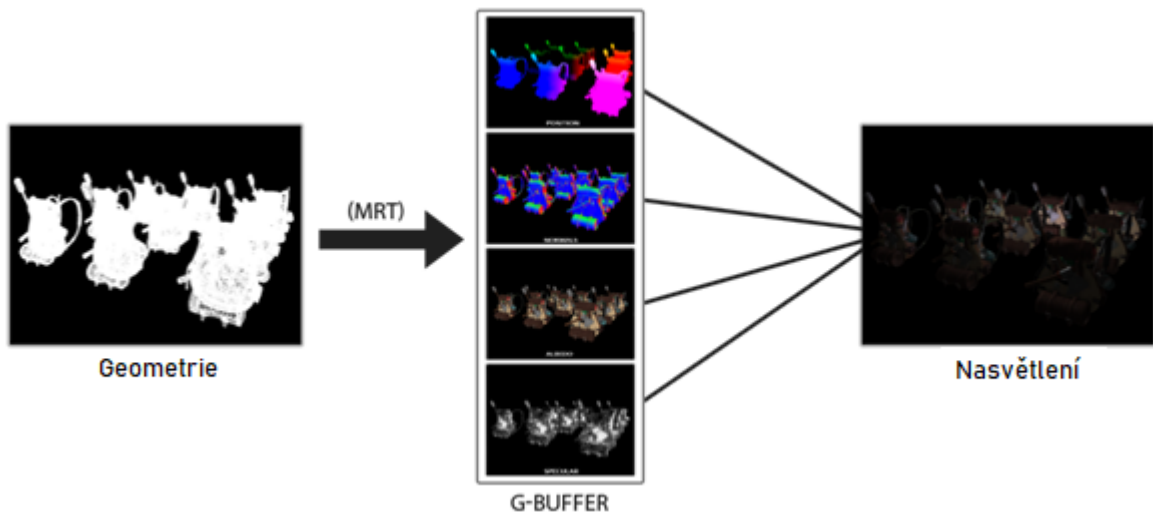
zatímco hráči očekávají od nových titulů lepší a lepší vizuální kvalitu. Proto musí renderovací enginy být vysoce optimalizované s konfigurovatelnou vizuální kvalitou během hraní.

V herní scéně jsou objekty, na určitých pozicích a kamera, která je snímá. Kamera má konfiguraci podobnou objektivu kamery ve skutečném světě. Nejdůležitějším parametrem je úhel pohledu, který je obvykle nastaven mezi 60 až 90 stupni, což odpovídá objektivu 18–28 mm na plném formátu. Blízká a vzdálená rovina definují vzdálenost od kamery, kde jsou objekty viditelné. S těmito informacemi může herní engine vypočítat matici pro transformaci objektů z globálních souřadnic do pohledu kamery. Pro efektivnější práci se zdroji se používají metody pro optimalizaci. Mezi takovouto metodu patří proces occlusion culling, ten odřízne všechny objekty mimo zorné pole (viz. obrázek 3). Zbývající objekty jsou seřazeny podle hloubky a renderovány. (Lee, 2023)



Obrázek 3 Vizualizace occlusion culling (Lee, 2023)

Existují různé přístupy k renderování objektů. Tradiční Forward rendering počítá osvětlení a materiály pro každou viditelnou geometrii a následně řeší, který z nich je nejbližší kamery, a ten je zobrazen. Aktuální přístup, který používají Unity a Unreal, se nazývá Deferred rendering. Geometrie je renderována na více průchodů (viz. obrázek 4). Stínování je následně spočítáno na základě těchto průchodů. Stínované jsou pouze viditelné fragmenty. Deferred rendering dokáže efektivně zpracovávat velké množství světel díky tomu, že počítá pouze stíny viditelných pixelů. Nevýhodou této techniky je že neumožňuje renderovat průhledné objekty a nepodporuje AASS (Anti aliasing - technika vyhlazování hran). (De Vries, 2024)



Obrázek 4 Znáznornění Deferred rendering přístupu (De Vries, 2024)

Herní engine obvykle poskytuje materiály nebo způsoby jejich vytváření a konfigurace. Materiály mohou být neosvětlené (jen barva, světla nemají vliv), osvětlené (stínován podle světla), průhledný nebo průsvitný. Textury jsou obrazy, které pomáhají definovat materiál. Existují různé druhy textur, které popisují barvu, drsnost, odlesk nebo posunutí materiálu.

Když je scéna renderována, engine aplikuje efekty post-processingu. Tyto efekty zlepšují vizuální kvalitu konečného výsledku. Nejčastějšími efekty jsou antialiasing (vyhlazuje hrany), ambientní occlusion (metoda stínování) a korekce barev konečného obrazu. (Unity Technologies, 2024)

### 3.2.2 Animace

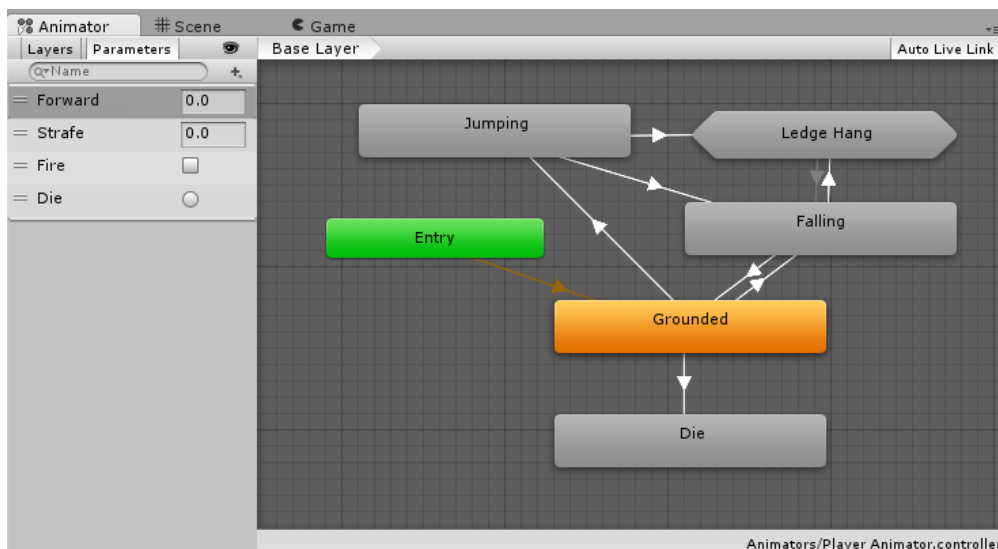
Ne všechny objekty ve scéně jsou statické. Objekty mohou být pohybovány fyzikou nebo animovány. Téměř ve každé hře je zapotřebí využít nějaké animace. Animací je považována za předdefinovanou změnu určitých parametrů v průběhu času. Může být animována pozice; celé postavy mohou být animovány pomocí kostí nebo vrcholy uvnitř sítě mohou být tvarovány do různých pozic a animovány. Jednoduché animace jsou obvykle vytvářeny přímo uvnitř herního engine. Například v situaci, kde hráč chce otevřít dveře. Je spouštěna animace otevírání, když se hráč přiblížil ke dveřím a stisknul akční tlačítko. Animace otevírání dveří je změna ve rotaci. Protože každé dveře se mohou otevírat odlišným způsobem, není žádané je animovat v 3D softwaru; je vhodnější animaci vytvořit přímo

v herním enginu. Herní enginy nabízejí prostředí pro animaci. Obvykle toto prostředí vypadá obsahuje graf. Na ose X je čas a na ose Y je hodnota, kterou chceme animovat. Může to být libovolná možná hodnota, kterou lze nastavit v enginu. Obvykle by mělo být možné změnit tvar křivky pro úpravu pohybu. (H. Eberly, 2005)

Komplexnější jsou animace koster. Aby byla postava uvedena do pohybu, musí mít rig. Rig je v podstatě kostra uvnitř modelu a propojení kostí s určitými částmi modelu. Například kosti uvnitř ruky skutečně pohybují prsty na ruce. Rigované postavy mohou být animovány ručně nebo pomocí techniky motion capture. Výsledkem tohoto procesu je například animace chůze. (BINARY MOON, 2024)

Herní engine nemusí poskytovat prostředí pro vytváření kosterních animací. Měl by však být schopen s nimi pracovat. Engine by měl být schopen přehrávat a míchat tyto animace dohromady tak, aby vytvářely plynulý pohyb. Hráč pak nepostřehne diskrétní přehrávání animací, ale vnímá postavu jako živou bytost, která může vykonávat různé úkoly. Pro ovládání animací postavy jsou k dispozici stavové automaty, které rozhodují, v jakém stavu se postava momentálně nachází a jakou animaci přehrát. Obrázek 5 představuje graf stavového automatu v herním engine Unity. Systémy mohou míchat více animací dohromady a upravovat je pro dosažení co nejplynulejších výsledků. (H. Eberly, 2005)

Nakonec lze také animovat tvar objektu. To je užitečné například pro animování výrazů obličejů. Různé pozice jsou importovány s objektem do enginu. Jedna z pozic je vždy základní pozice. Ostatní pozice jsou zpracovány jako relativní zkreslení hlavní pozice. Pokud je zkreslení nastaveno na jedničku, postava je plně v následující pozici. Tato nastavení lze animovat stejným způsobem jako ostatní. V Unity se tato technika nazývá Blend Shapes, v Unreal Morph Targets. (Epic Games, 2024; Unity Technologies, 2024)



Obrázek 5 Unity editor stavů animací a přechodů mezi nimi (Unity Technologies, 2024)

### 3.2.3 Fyzikální engine

Fyzika je důležitou součástí herního mechanismu. Většina her se odehrává ve skutečném světě, takže objekty ve scénách by se měly chovat fyzikálně správným způsobem. Herní engine obvykle poskytuje řešení pro takové fyzikální simulace. Každý objekt má colider (zjednodušená hranice objektu používaná pro fyziku) a nějaký fyzikální materiál, který určuje hmotnost a vlastnosti povrchu. Engine vypočítává vektory pohybu a kolize mezi objekty. Rozlišováno je mezi statickými objekty, které se vůbec nepohybují, jako je zem, a tělesy s hmotností, materiálem s třením, které reagují na síly, padají s gravitací atd. Navíc může engine simulovat měkké objekty, které mění svůj tvar v souladu s vnějšími silami (například látky). Ačkoliv fyzikální simulace nemusí být extrémně přesná, je důležitou součástí vývoje her. Unity a Unreal Engine využívají externí knihovnu pro fyziku PhysX založený na Nvidia GPU. (Unity Technologies, 2024)

### 3.2.4 Umělá inteligence

Kromě hráčova postavy videohry obvykle obsahují nepřátelské postavy, proti kterým se bojuje. Tyto postavy ovládané umělou inteligencí – potřebují určitou úroveň inteligence, aby se chovaly rozumně. Především u offline her pro jednoho hráče je potřeba, aby umělá inteligence byla trochu chytrá, aby hra byla zábavná. V kontextu her nejde zpravidla

o inteligenci jako takou spíše o její co nejuvěrnější napodobení. Cílem je vytvořit přesvědčivou iluzi.

Hlavní úkoly inteligence herních postav zahrnují nalezení cesty od bodu A do bodu B a chůzi s postavou podél této cesty. Herní engine poskytují řešení pro tyto úkoly. Obvykle engine generuje navigační plochu nazývanou NavMesh, která vychází ze statických objektů ve scéně. Tato síť může být během hry dynamicky upravována podle pohybu postav nebo fyziky tělesa s hmotností. Postavy ovládané naprogramovanou inteligencí jsou schopny chodit po této síti a hledat po ní cesty. Síť lze rozdělit do oblastí s různými náklady na pohyb, například cesty, bláto, bažiny, voda. Podle oblasti se může automaticky volit vhodná animace.

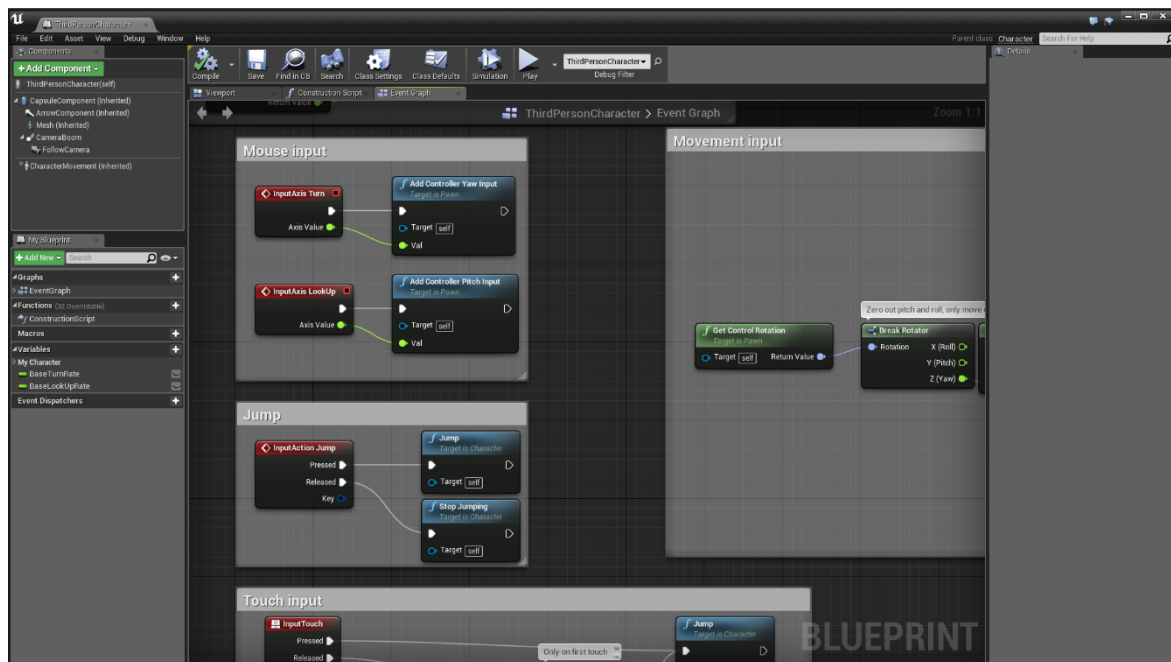
K vytvoření přesvědčivé umělé inteligence se nejběžněji používají stavové automaty, kde přechody mezi jednotlivými stavy jsou dané podle určitých pravidel. V Unreal Engine jsou implementovány stromy chování (anglicky behavioral trees). Tyto stromy jsou grafy, které jsou použity při rozhodování, jaký úkol by postava měla vykonávat. Nicméně herní engine nenahradí programování komplexnější umělé inteligence vývojáři. Nabízí pouze sadu základních nástrojů a systémů, které mohou využít k tomu, aby vykazovali chování podle požadovaných potřeb. (Apperson, 2023; Epic Games, 2024)

### 3.2.5 Scriptování

Herní engine musí také umožnit definovat chování jednotlivých herních komponent. To můžeme skripty, které jsou z pohledu herního engine komponenty pro objekty ve scéně. Ty pak umožňují programovat reakce na chování hráče a vzájemnou spolupráci mezi entitami.

V Unity je podporovaným programovacím jazykem C#. Unreal Engine nabízí nativní podporu programovacího jazyku C++ nebo vizuální skriptování pomocí Blueprint. Je důležité, aby skripty mohly být přidávány a znovu používány na objektech ve scéně jednoduchým způsobem, aby je mohl použít i herní designer bez znalosti programování. Jak daný engine pracuje s tímto problémem, závisí na editoru daného engine. V Unity může herní designer přidat skript jako komponentu objektu a následně lze upravovat jednotlivé atributy. V Unrealu existuje koncept Blueprint, což je vizuální programování, které

umožňuje uživateli vytvářet složité logiky bez napsání jediného řádku kódu. (Epic Games, 2024; Sewell, 2015)



Obrázek 6 Blueprint editor v Unreal Engine (Sewell, 2015)

### 3.2.6 Zvuk

Důležitou součástí každé hry je zvukový design a hudba, která vytváří atmosféru. Herní enginey poskytují nástroje pro přehrávání a zastavování soundtracků na základě událostí ve hře. Pokročilé zvukové enginey mohou simulovat ozvěnu ve 3D prostoru nebo přehrávat prostorový zvuk podle pozice hráče.



### 3.3 Unity

Unity je herní engine umožňující vývoj širokého množství aplikací pro mnoho platforem. V dnešní době je nejvyužívanějším herním enginem jeho podíl na trhu je kolem 40 % (Sawant, 2023). Díky tomu, jak je široce využíván je k dispozici široké množství návodů a assetů. Hlavní myšlenkou Unity je uživatelsky přívětivý přístup a multiplatformní využití. To dělá Unity ideální volbou pro menší nezávislé vývojáře. (Piskunov, 2021)

#### 3.3.1 Historie

Unity bylo původně vyvinuto pro systém MacOS. V roce 2005 bylo poprvé představeno veřejnosti. Už od verze 1.1 ale umožňovalo exportovat hry na operační systém Microsoft Windows a do webových prohlížečů. Unity přineslo do webového prohlížeče hardwarově akcelerovanou grafiku a možnosti, které předtím s technologií flash nebyly možné.

V roce 2007 vznikla verze 2.0, která rozvinula podporu operačního systému Windows. Nově také podporovalo Microsoft DirectX a přineslo spoustu dalších funkcí.

Již z počátku propagovalo Unity myšlenku multiplatformního vývoje. Pod vydání iPhone se rozhodli vydat verzi Unity iPhone, která umožnila vyvíjet na hry mobilní telefony.

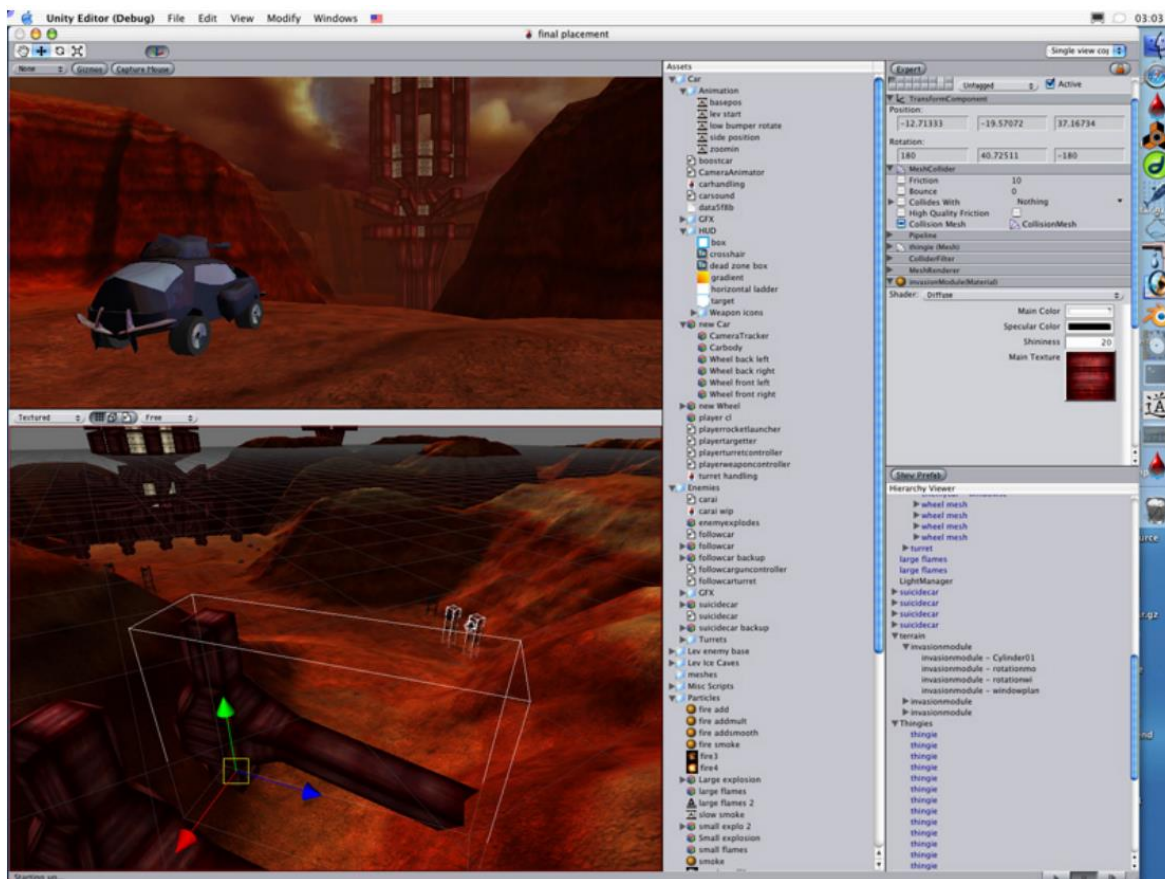
První verze Unity, která byla spustitelná na Windows byla verze 2.5 vydaná v roce 2009. Toto vyžadovalo kompletně přepsat celý engine a udělat kód nezávislý na platformě.

Dalším významným vydáním byla verze 3.0. Přinesla spoustu žádaných funkcí, mezi které patří například deferred rendering, occlusion culling nebo debugging a nižší úrovní. Editor Unity iPhone a Unity Wii byl sjednocen v jeden jediný multiplatformní editor. S touto verzí Unity přesáhlo dvě stě tisíc uživatelů a stalo se nejpoužívanější technologií pro vývoj mobilních aplikací.

S dalším velkým vydáním 4.0 Přišla do Unity podpora pro vydávání her pro Linux a Adobe Flash. Ve verzi 4.3 také přišla nativní podpora vývoje dvou dimenzionálních her. Vznamnou novinkou byl také animační systém Mecanim.

V roce 2015 byla vydána verze 5.0. Tato verze přinesla mnoho novinek. Vývoj webových her s využitím WebGL nově nevyžaduje instalaci pluginu na stroji uživatele. Unity Cloud umožňuje zprávu a sdílení projektů mezi vývojáři. Přicházení nové světelné a částicové efekty, globální iluminace, podpora Vulkan API, vývoj na platformu Nintendo

Switch a podporu Nvidia PhysX fyzikálního enginu. Od roku 2017 se změnilo číslování verzí a nadále začali vycházet nové verze s větší kadencí. Nově vychází pravidelně každé čtvrtletí nová verze s aktuálními vylepšeními. (Haas, 2020; Academic Accelerator, 2024).



Obrázek 7 Screenshot Unity verze 0.2b (Haas, 2020)

### 3.3.2 Cena

Společnost Unity Technologies nabízí cenový model založený na ročním předplatném. Toto předplatné je placeno formou pravidelných poplatků za jednotlivé uživatele. K dispozici jsou různé úrovně předplatného, včetně bezplatné verze s omezenými funkcemi. Ta je určena pro hobby vývojáře, studenty, nebo začínající studia. Placenou verze představuje verze Pro, ta vyžaduje měsíční nebo roční předplatné a nabízí další funkce a výhody.

Výhodami plynoucími z placené verze licence je přístup k nástrojům Unity Cloud, Mars, Havok Physics. Dále také možnost publikace her na konzole, nebo odstranění loga Unity, které se zobrazuje při spuštění aplikace. Unity Cloud je nástroj pro sdílení a projektu.

Unity Mars je sada nástrojů pro usnadnění vývoje pro AR/VR. Placená verze také zajišťuje prioritní zákaznickou podporu.

Kromě licence Pro nabízí Unity také alternativní možnosti licencování pro větší studia a projekty na podnikové úrovni. V těchto případech je cena individuálně sjednána pro s konkrétními společnostmi. Tato cenová struktura umožňuje vývojářům zvolit si úroveň funkcí a podpory, která nejlépe vyhovuje jejich rozpočtu a požadavkům projektu. (Piskunov, 2021; Unity Technologies, 2024)

Bezplatná verze je omezena na využití pouze společnostmi s příjmem do dvou set tisíc dolarů za kalendářní rok. Ceny placených předplatných se pohybují od 1877 dolarů do 4554 dolarů za jednu licenci na rok. (Unity Technologies, 2024). Pokud příjmy z jedné vydané hry dosáhnou výše jednoho miliónu dolarů a zároveň také jednoho miliónu stažení, vyžaduje navíc Unity placení takzvaného „Unity Runtime Fee“. Výše tohoto poplatku je závislá na měsíční výši příjmu a počtu instalací hry. Nikdy však nepřesáhne 2,5 % ročního příjmu. (Unity Technologies, 2024)

### **3.3.3 Budoucnost**

V nadcházející verzi Unity 6, plánuje společnost přechod k AI nástrojům. Inteligentní asistent Unity Muse bude asistovat při psaní kódu, generování jednoduchých 2D assetů a také vytváření textur. V dalších verzích bude Unity Muse také umět vytvářet animace a v neposlední řadě nabídne nástroje pro vybavení herních postav a charakterů umělou inteligencí. To umožní realističtější zážitky z her a usnadní vývoj. (Griffiths, 2023)

### **3.3.4 Dostupnost návodů a assetů**

Unity je nejpoužívanější herní engine na trhu, a to má mimo jiné za efekt lepší dostupnost různých materiálů, pluginů a podporu širokého množství programů. Unity nabízí platformu Unity Learn, které obsahuje široké množství návodů a tutoriálu pro různá využití herního enginu. (Piskunov, 2021)

Za konkurenční výhodu Unity lze také považovat rozsáhlý obchod s assety Unity Asset Store. Ten nabízí velmi široké množství stažitelných assetů od modelů, textur po komplexní nástroje, které usnadní vývoj rozsáhlejších her.

### 3.3.5 Příklady her

Unity je používáno největšími herními studii současnosti, například Blizzard Entertainment používá Unity pro vývoj jejich titulu Hearthstone (Unity Technologies, 2014). Ta je ukázkou nejsilnější stránky Unity a to možností přenášet jeden produkt mezi více platformem. Obecně lze říci že největší projekty z pohledu počtu uživatel jsou v dnešní době na mobilních platformách, a právě tam Unity dominuje. (Reed, 2023)

### 3.3.6 Kompatibilita s různými platformami

Unity nabízí kompatibilitu s velkým množstvím platformem. Mezi tyto platformy patří:

- Windows
- Mac
- Linux
- iOS
- Android
- Xbox
- PlayStation
- Nintendo Switch
- Google Stadia
- WebGL

Velké množství podporovaných platformem dává Unity konkurenční výhodu oproti ostatním herním enginům na trhu. (Piskunov, 2021)

### 3.3.7 Programovací jazyky

Primární programovací jazyk v herním enginu Unity jazyk C#. K psaní skriptů je nutné využít externí vývojové prostředí, jako například Visual Studio od společnosti Microsoft.

S využitím pluginů, lze Unity využít i jiné programovací jazyky, jako například Python. Nebo také programovat vizuálně s podobnými nástroji jako Blueprint v Unreal Engine. Tyto pluginy, ale nejsou přímo podporovány a vyvíjeny Unity. (Piskunov, 2021)

## 3.4 Unreal Engine

Unreal Engine od společnosti Epic Game je multiplatformní herní engine. Je druhý nejpoužívanější herní engine. Jeho podíl na trhu herního vývoje je přibližně 30 % (Sawant, 2023). Unreal Engine se orientuje převážně na vysoce detailní 3D hry. Kvalita a rychlost renderování představuje jeho největší konkurenční výhodu.

Mimo svět her je Unreal Engine také využíván ve filmovém průmyslu. V sérii Mandalorian je dynamicky simulováno pozadí mnoha scén v Unreal Engine 5. Podobně byl Unreal Engine využit v sérii Westworld od HBO a mnoho dalších studií uvažuje o jeho využití. (Jensen, 2024)

### 3.4.1 Historie

V devadesátých letech minulého století byl ve vývoji her kritický přechod z dvou prostorových her na tři dimenzionální polygonovou grafiku. Přesně v této problematice uměla využít tehdy malá firma Epic Games. V roce 1998 byla vydána první hra Unreal vyvinutá za pomoci Unreal Engine verze 1. Po úspěchu Unrealu přešla na Unreal Engine jiná studia, nejznámějším příkladem je Legend Entertainment a Micropose vyvíjející Duke Nukem. Epic vybudoval robustní editor a programovací jazyk UnrealScript, který umožňoval hobby programátorům a hráčům, vytvářet modifikace her.

Unreal Engine 2 byla první verze která umožňovala vyvíjet hry pro herní konzole, vydána byla v roce 2002. Unreal Engine 2 také s novými funkcemi umožňoval vytvářet jiné žánry her než pouze střílečky.

V roce 2006 byl vydán Unreal Engine 3, který přinesl podporu vývoje pro systémy iOS a Android. Přinesl také vylepšený rendering a fyzikální výpočty. V roce 2009 byl vydán Unreal Development Kit, který dal přístup hobby a indie vývojářům k Unreal Enginu.

Dalším obrovským krokem kupředu byla verze 4. Vydaná byla v roce 2014. Poprvé byl představen skriptovací editor Blueprint, který umožnil vytvářet skripty bez nutnosti znalosti programovacího jazyku C++. Byl vydán Unreal Marketplace, na kterém mohou uživatelé prodávat a nakupovat assety, kód nebo různé nástroje.

Mezi jednoznačně největší úspěchy společnosti Epic Games patří titul Fortnite vydaný v roce 2017. Po vydání nesklidila tato hra příliš úspěchů. Zachránil ji ale nápad vytvořit jednoduchý a zdarma přístupný mód „battle-royale“. Z tohoto módu se vznikla

jedna z nejúspěšnějších her historie herního vývoje. Přinesla Epic Games přes deset biliónů dolarů. S Unreal verzí 5 byla vydán editor Unreal Engine for Fortnite, který umožňuje vytváření a modifikaci originální hry. (Jensen, 2024; Salama, 2021).

### 3.4.2 Cena

Epic Game nabízí tři možnosti licencování jejich herního enginu Unreal Engine:

- Standard
- Enterprise
- Custom

Standartní licence Unreal Engine je k dispozici zadarmo. Je určena především pro začínající herní vývojáře a jiné menší společnosti. Licence Enterprise je určena pro neherní společnosti, které vyžadují větší míru podpory od Epic Games. Její součástí je kromě prioritní zákaznické podpory například možnost vzdělávání uživatelů oficiálními kurzy. Licence Custom je pak určena pro ty největší herní studia a pro jiné specifické případy, ve kterých se neaplikuje požadavek na placení poplatku z příjmů. Mezi tyto případy patří neherní využití enginu například ve filmovém průmyslu nebo architektuře. (Epic Games, 2024)

*Tabulka 1 Licenční poplatky za využívání Unreal Engine (Epic Games, 2024)*

<b>Příjem z vydané aplikace</b>	<b>Poplatek</b>
Méně jak 1 milión USD	0 USD
Více než 1 milión USD	5 % z příjmu

Pro všechna využití je Unreal Engine zadarmo, pokud vydaná aplikace nedosáhne příjmu většího jak jeden milión dolarů za dobu své existence. Pokud tuto hodnotu přesáhne, musí autoři platit pět procent z příjmů získaných prostřednictvím konkrétní aplikace za rok. Poplatky jsou placeny čtvrtletně. Toto platí, dokud aplikace vydělává alespoň deset tisíc dolarů za čtvrtletí. (Filmmaker, 2023)

V roce 2024 by měl Unreal Engine využívaný mimo herní průmysl přejít na roční poplatky za licenci podobně jako u Unity. V oblasti herního vývoje ovšem zůstane stejný finanční model. (Sato, 2023)

### **3.4.3 Budoucnost**

S verzí Unreal Engine 5 představilo Epic Games revoluční přístup k renderování pomocí technologie Nanite. Tato technologie umožňuje renderovat velice detailní modely bez nutnosti vytváření LOD (level of detail) modelů.

Podobně revoluční je technologie ray tracingu, v podání Unreal Engine se nazývá lumen. Obě tyto technologie, Nanite a Lumen, dnes představují vrchol toho, co je v herním vývoji možné z pohledu kvality a detailu grafických vizuálů dosáhnout.

Mezi další novinky, na které Unreal Engine přinese budou patřit nástroje pro procedurální generování prostředí. Také větší optimalizace enginu pro mobilní zařízení. (Tiltedu, 2023).

### **3.4.4 Dostupnost návodů a assetů**

Unreal Engine nabízí širokou nabídku oficiálních návodů, které jsou součástí dokumentace samotného enginu. Navíc je také dostupné forum Epic Developer Community, které je prostorem pro komunikaci s ostatními vývojáři ohledně všech možných témat souvisejících s vývojem aplikací v Unreal Engine. Kromě oficiálních platform, podobně jako u Unity, díky velikosti komunity lidí využívajících Unreal Engine, existuje velké množství neoficiálních návodů a tutoriálů.

Unreal Engine Marketplace představuje klíčový zdroj pro vývojáře, kteří pracují s Unreal Enginem, nabízející širokou škálu aktiv, zdrojových kódů a nástrojů pro optimalizaci vývoje her. Tato platforma umožňuje vývojářům zakoupit a integrovat vysoce kvalitní assety přímo do svých projektů, což výrazně zrychluje proces vývoje a umožňuje jim dosáhnout profesionálního vzhledu a funkcionality bez nutnosti vytvářet všechno od nuly. Díky širokému spektru dostupných assetů, od 3D modelů a textur po animace a zvukové efekty, mají vývojáři možnost efektivně rozšířit své projekty a přizpůsobit je svým potřebám. To nejenže snižuje čas potřebný k vývoji, ale také poskytuje inspiraci a možnost učení se od zkušenějších tvůrců v komunitě.

### 3.4.5 Příklady her

Unreal Engine byl použit při vývoji mnoha úspěšných titulů, které ovlivnily herní průmysl a poskytly hráčům nezapomenutelné zážitky. Následující jsou příklady několika populárních her vyvinutých pomocí Unreal Engine:

- **Fortnite od Epic Games, 2017** - Fortnite je multiplayerová battle royale hra, která se stala jedním z fenoménů současného herního světa. Vyvinuta společností Epic Games, využívá Unreal Engine pro svou vynikající grafiku, dynamické prostředí a kreativní hratelnost. Fortnite získal obrovskou popularitu díky svému jedinečnému vizuálnímu stylu, sezonním aktualizacím a širokému spektru herních režimů.
- **Gears of War série od Epic Games a The Coalition – Série Gears of War** patří k nejuznávanějším třetí osobou střílečkám. Vyvinutá původně společností Epic Games a později The Coalition, tato série využívá Unreal Engine pro dosažení plynulé akce, detailních postav a impozantních prostředí. Vývojáři využili Unreal Engine k vytvoření ikonického vizuálního stylu a inovativních herních prvků.
- **Unreal Tournament od Epic Games, 1999** - Hra, která dala jméno celé hernímu enginu, Unreal Tournament, byla původně vyvinuta samotnou společností Epic Games. Tato multiplayerová střílečka získala uznání díky své akci, grafice a možnostem modifikací, což vše bylo umožněno díky Unreal Engine.
- **Street Fighter V od Capcom, 2016** – I když je Unreal Engine spíše spojen s akčními a adventurními hrami, Street Fighter V od společnosti Capcom ukazuje jeho všestrannost. Tato bojová hra využívá Unreal Engine pro vytvoření plynulých animací, detailních bojovníků a realistických arén.
- **PUBG od PUBG Corporation, 2017** - PlayerUnknown's Battlegrounds, známé též jako PUBG, bylo jedno z prvních velkých multiplayerových her žánru battle royale. Unreal Engine pomohl vytvořit rozlehlé herní prostředí, realistickou grafiku a plynulou hratelnost, což přispělo k úspěchu této hry.

Tyto příklady jen zdůrazňují pestrost projektů, které byly úspěšně vytvořeny s využitím Unreal Engine. Engine poskytuje vývojářům nástroje a prostředky pro vytváření her napříč různými žánry, což z něj činí jednu z předních platforem pro tvorbu herních zážitků. (Porokh, 2023).



### 3.4.6 Kompatibilita s platformami

Unreal Engine se vyznačuje svou schopností poskytovat multiplatformní podporu, což znamená, že vývojáři mají možnost vytvářet hry a aplikace pro různé platformy. Tato flexibilita přináší několik klíčových výhod, včetně rozšíření publika, maximalizace dosaženého zisku a možnosti vytvářet komplexní zážitky pro různé zařízení.

Platformy podporované Unreal Engine:

- **PC (Windows, Linux, macOS):** Unreal Engine poskytuje robustní nástroje pro vývoj her na osobní počítače s různými operačními systémy. To zahrnuje Windows, Linux a macOS, což umožňuje vývojářům cílit na širokou škálu uživatelů.
- **Konzole (PlayStation, Xbox, Nintendo):** Unreal Engine nabízí plnou podporu pro vývoj her na herní konzole, včetně PlayStation (PS4, PS5), Xbox (Xbox One, Xbox Series X/S) a platformy Nintendo Switch.
- **Mobilní zařízení (iOS, Android):** Vývoj pro mobilní zařízení, jako jsou telefony a tablety, je také podporován. Unreal Engine umožňuje vytvářet vysokokvalitní hry pro iOS a Android.
- **Virtual reality (VR):** Pro vývojáře, kteří se zaměřují na virtuální realitu, Unreal Engine nabízí podporu pro VR platformy, včetně HTC Vive, Oculus Rift a dalších.
- **Augmented reality (AR):** Unreal Engine podporuje vývoj aplikací s rozšířenou realitou pro zařízení jako je Microsoft HoloLens a mobilní telefony s AR funkcionalitou. (Epic Games, 2024).

### 3.4.7 Programovací jazyky

Unreal Engine poskytuje vývojářům rozsáhlou paletu nástrojů pro implementaci funkcí a interakcí v jejich herních projektech. Hlavním programovacím jazykem v Unreal Engine je C++, který tvoří základní strukturu engine a poskytuje vývojářům plný přístup k enginovému API. Tím umožňuje detailní kontrolu nad funkcionalitou hry, implementaci herní logiky a optimalizace kódu pro dosažení optimálního výkonu.

C++ v Unreal Engine slouží k tvorbě vlastních tříd, funkcí a modulů, což vývojářům umožňuje flexibilně přizpůsobit engine specifickým potřebám jejich projektů. Tento jazyk je vhodný pro pokročilý vývoj a optimalizaci herních systémů.

Unikátním prvkem Unreal Engine je však také programovací prostředí známé jako Blueprint. Jedná se o vizuální programovací jazyk, který umožňuje vývojářům vytvářet logiku pomocí grafických bloků, které jsou propojovány a definují chování a interakce ve hře. Blueprint se zaměřuje na jednoduchost použití a intuitivní přístup k vytváření herních prvků bez potřeby hlubších znalostí programování.

Blueprint je ideální pro rychlé prototypování a iterativní vývoj, což umožňuje vývojářům snadno definovat herní logiku, animace, umělou inteligenci a události. Jeho drag-and-drop přístup a vizuální reprezentace logiky zvyšují přístupnost pro ty, kteří nemají rozsáhlé programátorské dovednosti.

Unreal Engine podporuje hybridní přístup, kdy lze kombinovat jak C++, tak Blueprint v jednom projektu. Tato kombinace umožňuje vývojářům využít výhody obou prostředí a optimalizovat vývoj dle specifických požadavků jejich projektů. C++ je vhodné pro náročné výpočty a optimalizace, zatímco Blueprint nabízí rychlý vývoj a iterativní úpravy, což společně tvoří efektivní prostředí pro tvorbu výkonných a vizuálně poutavých her. (Piskunov, 2021; Epic Games, 2024).

## 3.5 Tvorba Prototypů

### 3.5.1 Co je to prototyp

První fází vývoje herní aplikace bývá vytvoření prototypu. V této kapitole jsou představeny druhy prototypů a následně je porovnán přístup k tvorbě prototypu v Unity a Unreal Engine.

Prototyp ve vývoji her představuje ranou verzi hry, která je vytvořena s cílem ověřit a demonstrovat klíčové koncepty, herní mechaniky a návrh interaktivních prvků. Tento koncept je zásadní pro iterativní proces vývoje, umožňuje vývojářům a designérům testovat a zlepšovat nápady před tím, než se pustí do drahého a časově náročného vývoje plnohodnotné verze hry. Prototypování poskytuje cennou zpětnou vazbu od uživatelů a stakeholderů v rané fázi vývojového cyklu, což umožňuje identifikaci a řešení potenciálních problémů a nedostatků. Díky prototypům lze také lépe odhadnout rozsah projektu, jeho technické požadavky a celkovou životaschopnost. V praxi mohou být použity různé úrovně prototypů, od jednoduchých „papírových“ modelů po interaktivní digitální verze, které simulují skutečnou herní zkušenost. Tento přístup podporuje agilní metodologie vývoje, kde je kladen důraz na rychlé iterace, flexibilitu a adaptabilitu. (Schell, 2008)

Typy prototypů ve vývoji počítačových a mobilních her lze kategorizovat do několika základních skupin, každý s různým zaměřením a účelem v rámci vývojového procesu.

1. **Koncepční prototypy** – Tyto prototypy jsou často velmi abstraktní a zaměřují se na testování základních myšlenek nebo konceptů hry. Mohou být realizovány i mimo digitální prostředí, například pomocí papírových prototypů.
2. **Hratelné prototypy** – Slouží k otestování a vylepšení herních mechanik a interakce. Hratelné prototypy jsou často vytvářeny s minimálními grafickými a audiovizuálními prvky, aby se vývojáři mohli soustředit na základní herní zážitky.
3. **Vizuální prototypy** – Tyto prototypy se zaměřují na grafický design, vizuální styl a atmosféru hry. Pomáhají týmu lépe pochopit, jak by hra měla vizuálně vypadat, ale nemusí nutně obsahovat hratelné prvky.
4. **Technické prototypy** – Slouží k testování specifických technologických výzev, jako jsou výkonnostní testy, integrace třetích stran (např. API pro sociální sítě nebo platební systémy), nebo testování nových technologií (např. VR nebo AR).

5. **Prezentační prototypy** – Tento typ prototypu je navržen pro prezentaci hru investorům, vydavatelům, nebo jiným zainteresovaným stranám. Jsou navrženy tak aby udělaly dobrý dojem a ukázaly potenciál plné verze hry. (Schell, 2008).

### 3.5.2 Tvorba prototypu v Unity

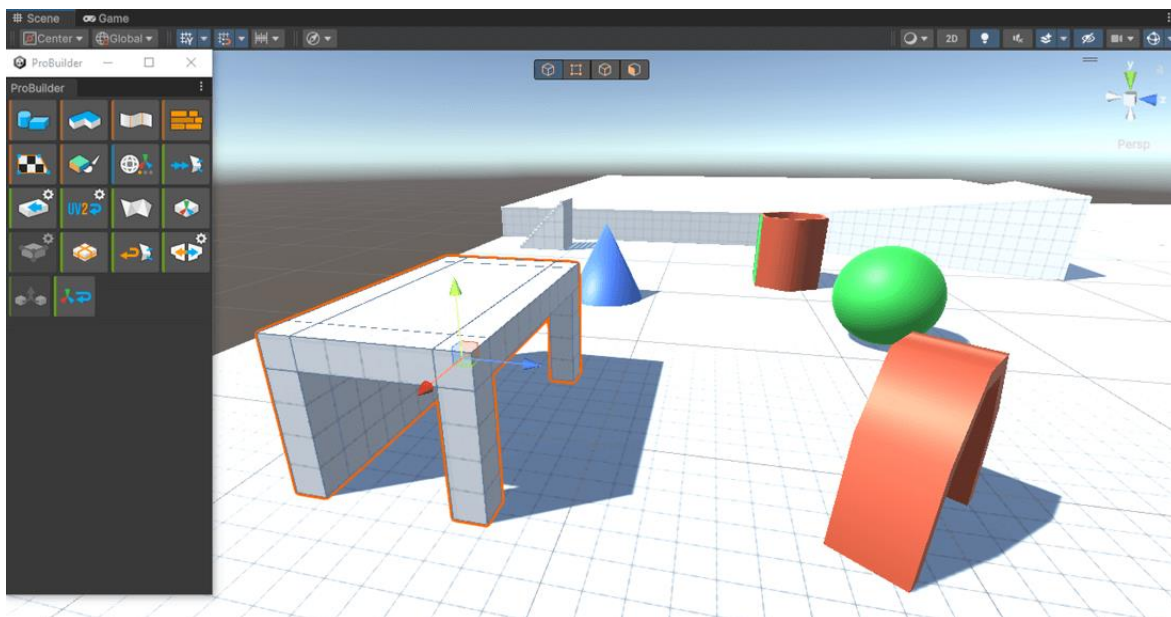
Při vytváření prototypu se často začíná s vytvořením jednoduchých úrovní pomocí techniky „whiteboxing“, kde se používají základní geometrické objekty k definování prostoru a herních prvků. Následně se přidávají umělecké, zvukové a osvětlovací prvky, aby se úroveň oživila. Kódování v C# je nezbytné pro přidání funkcionalit a odezvy hráče, což umožňuje vytvořit hratelný prototyp.

ProBuilder já nástroj umožňující tvůrcům rychle modelovat a prototypovat 3D objekty a prostředí přímo v editoru Unity. Uživatelské prostředí toho nástroje je vidět na obrázku 8. Tento nástroj je ideální pro návrh úrovní a experimentování s geometrickými tvary, což usnadňuje iterativní proces vývoje her. Umožňuje jednoduché vytváření a manipulaci s objekty, jako jsou bloky, válce a sféry, které lze použít pro rychlé sestavování herních scén.

Pokročilé funkce ProBuilderu zahrnují úpravu UV, práci s vertexovými barvami a vytváření parametrických tvarů. Tyto nástroje poskytují větší kontrolu nad vzhledem a pocitem modelovaných objektů. Umožňují také efektivní práci s texturami a materiály, což je zásadní pro vytvoření vizuálně přitažlivých scén.

Export modelů z ProBuilderu do externích 3D modelovacích programů rozšiřuje možnosti úprav a detailů. Tato funkce je užitečná pro jemné doladění modelů nebo pro integraci práce provedené v jiných nástrojích zpět do Unity projektu, což umožňuje plynulý pracovní postup mezi různými platformami.

Pokud jde o základní 3D objekty, v Unity je k dispozici sada předdefinovaných primitiv, které lze snadno vložit a upravit v scéně. Tyto objekty mohou sloužit jako výchozí body pro tvorbu složitějších tvarů a jsou základem pro prototypování a návrh herních úrovní. ProBuilder dále rozšiřuje možnosti práce s těmito primitivy, poskytující tvůrcům komplexní sadu nástrojů pro rychlou a efektivní tvorbu 3D obsahu. (Unity Technologies, 2024; Nobreakpoints, 2024)



Obrázek 8 Editor Probuilder otevřený v Unity (Nobreakpoints, 2024)

### 3.5.3 Tvorba prototypu v Unreal Engine

Tvorba prototypů her v Unreal Engine je proces, který umožňuje vývojářům rychle experimentovat s herními mechanikami, designem úrovní a vizuálními efekty. Tento proces je základem pro testování a iteraci nápadů, které mohou být rozvinuty do plně funkčních her. Unreal Engine, díky své robustní sadě nástrojů a pokročilým funkcím, poskytuje ideální prostředí pro efektivní prototypování. Klíčové nástroje, jako je Blueprint, Level Editor, a Material Editor, hrají zásadní roli v tomto procesu, umožňující vývojářům rychle realizovat a testovat své nápady.

Vizuální skriptování Blueprint je jedním z nejvýznamnějších nástrojů v Unreal Engine pro prototypování, který umožňuje vývojářům a designérům vytvářet komplexní herní logiku bez nutnosti psaní kódu. Díky vizuálnímu rozhraní a systému "drag-and-drop" mohou uživatelé snadno experimentovat s různými herními mechanikami a rychle implementovat změny. Tato vlastnost činí Blueprint neocenitelným nástrojem pro prototypování, jelikož umožňuje rychlou iteraci a testování bez hluboké technické expertízy v programování.

Level Editor v Unreal Engine nabízí další klíčovou výhodu pro tvorbu prototypů. Tento nástroj poskytuje intuitivní rozhraní pro design a organizaci herních světů, umožňující vývojářům snadno umístit a upravit objekty ve scéně. S funkcemi, jako je editor terénu a podpora dynamického osvětlení, mohou vývojáři rychle vytvořit vizuálně působivé

a funkční prototypy. To je zásadní pro testování designu hry a interakce hráče s herním prostředím.

Material Editor a systém částic jsou dalšími významnými nástroji v Unreal Engine, které umožňují vývojářům přidávat detaily a realističnost do svých prototypů. Material Editor nabízí rozsáhlé možnosti pro vytváření a úpravu materiálů a povrchů, což vývojářům umožňuje vytvářet vysoce kvalitní vizuální efekty. Systém částic pak podporuje vytváření složitých efektů, jako jsou oheň, kouř, nebo magické efekty, což je nezbytné pro dodání vizuálního dopadu a atmosféry herním prototypům.

Výsledkem je, že Unreal Engine nabízí vývojářům výkonné a flexibilní prostředí pro tvorbu prototypů her. Díky kombinaci Blueprint Visual Scripting, Level Editoru, Material Editoru a dalších nástrojů mohou vývojáři rychle a efektivně testovat a iterovat své nápady. Toto umožňuje nejen rychlou realizaci prototypů, ale také vysokou úroveň vizuální kvality a interaktivnosti, což je klíčové pro úspěšný vývoj her. (Epic Games, 2024)

### **3.5.4 Porovnání tvorby prototypů**

Tvorba prototypů je zásadní fází vývoje her a interaktivních aplikací, která umožňuje vývojářům testovat a iterovat koncepty rychle a efektivně. Unity a Unreal Engine jsou dvěma předními herními enginy, které se v této oblasti často porovnávají. Každý z nich má své specifické výhody a nevýhody, což vede k rozdílnému uplatnění v praxi.

Unity je často chváleno přehlednějším a snadněji použitelným editorem a také za to, že se dá rychleji naučit s ním pracovat, díky čemuž je ideální volbou pro začínající vývojáře a menší studia. Jeho skriptovací jazyk C# je považován za jednodušší na naučení ve srovnání s C++ používaným v Unreal Engine, což může urychlit prototypování pro nováčky v programování.

Na druhou stranu, Unreal Engine je vysoce ceněn pro své pokročilé grafické schopnosti a vizuální kvalitu, což jej činí preferovanou volbou pro projekty vyžadující vysokou úroveň detailů a realističnosti. Využití Blueprint Visual Scripting navíc umožňuje rychlé prototypování i pro ty, kteří nejsou zkušení programátoři.

Unity je oceňováno pro jeho všestrannost a schopnost pracovat na široké škále platform s minimálními úpravami. Tato flexibilita se může ukázat jako výhodná při prototypování aplikací určených pro mobilní zařízení nebo VR/AR systémy. Unreal Engine,

ačkoliv je také velmi škálovatelný, může vyžadovat více zdrojů a úprav pro optimalizaci na nižších systémech.

Oba enginy mají velmi aktivní a podpůrné komunity. Unity má výhodu v širší bázi uživatelů, což vede k většímu množství dostupných zdrojů, tutoriálů a assetů. Unreal Engine má sice menší, ale velmi odbornou komunitu, která se specializuje na graficky náročné a technologicky pokročilé projekty.

Unity a Unreal Engine se liší i v licenčních modelech. Unreal Engine nabízí svůj software zdarma, ale vyžaduje odvody z příjmů projektů, které přesáhnou určitý finanční limit. Unity nabízí flexibilnější licenční modely, včetně bezplatné verze s omezeními, která může být atraktivnější pro nezávislé vývojáře a malá studia.

V otázce výkonu a optimalizace, Unreal Engine může nabídnout lepší výsledky „out of the box“ pro projekty vyžadující intenzivní grafiku a fyziku, zatímco Unity může vyžadovat více manuální optimalizace pro dosažení srovnatelné vizuální kvality ve vysokých nastaveních.

V kontextu budoucích trendů a technologií, Unreal Engine se rychle adaptuje na nové průmyslové standardy, jako je ray tracing a virtuální produkce, což naznačuje jeho zaměření na špičkovou vizuální kvalitu a realističnost. Unity drží krok s důrazem na multiplatformní integraci a rozvoj nástrojů pro VR/AR, což otevírá široké možnosti pro různé typy projektů.

V konečném srovnání, výběr mezi Unity a Unreal Engine by měl být řízen specifickými potřebami projektu, zkušenostmi týmu a cílovou platformou. Unity se jeví jako výhodnější pro projekty s omezeným rozpočtem, rychlé prototypování a aplikace na více platformech, zatímco Unreal Engine vyniká v projektech vyžadujících pokročilé grafické a vizuální efekty, realistické simulace a špičkovou vizuální kvalitu.

## **3.6 Implementace online her pro více hráčů**

Tato kapitola se věnuje tvorbě online her pro více hráčů. Nejdříve je jsou představeny nástroje a přístup v Unity a Unreal Engine a následně jsou porovnány.

### **3.6.1 Multiplayerový vývoj a nástroje v Unity**

Implementace online her v Unity představuje komplexní proces, který vyžaduje hluboké porozumění síťové architektury, komunikace mezi klientem a serverem

a optimalizace výkonu pro zajištění plynulého multiplayerového zážitku. Unity nabízí vývojářům různé nástroje a služby, které usnadňují tvorbu online her, včetně Unity Networking (UNET), Photon Unity Networking (PUN) a Mirror. Tyto nástroje poskytují rámce pro vytváření síťových aplikací, zahrnující vše od základních síťových operací po komplexní systémy pro multiplayerové hry. (Photon, 2024)

Unity Networking (UNET) byl dlouho základním stavebním kamenem pro implementaci síťových funkcí v Unity, ačkoliv jeho vývoj byl pozastaven ve prospěch novějších řešení. UNET poskytoval vysokou úroveň abstrakce síťových operací, umožňující vývojářům snadno synchronizovat stavy objektů mezi klienty a serverem a implementovat volání vzdálených procedur (RPC). Přestože je UNET považován za zastaralý, jeho koncepty a implementační modely stále ovlivňují současné přístupy k síťovému programování v Unity.

Photon Unity Networking (PUN) je populární alternativou k UNET, která se stala výběrem mnoha vývojářů díky své škálovatelnosti a snadné integraci. PUN poskytuje cloudovou službu pro multiplayerové hry, což umožňuje snadné připojení hráčů z různých geografických oblastí s minimální latencí. PUN nabízí podporu pro matchmaking, synchronizaci stavů a volání RPC, což vývojářům usnadňuje vytváření robustních online her bez nutnosti podrobné znalosti síťové infrastruktury.

Mirror je moderní síťový engine pro Unity, který se zaměřuje na jednoduchost, výkon a flexibilitu. Jako open-source alternativa k UNET, Mirror usiluje o poskytnutí čistého a snadno použitelného API pro vývoj multiplayerových her. Mirror je navržen s důrazem na nízkou latenci a vysokou škálovatelnost, což umožňuje vývojářům vytvářet online hry od malých projektů až po hry s velkým počtem hráčů. Díky své komunitě a pravidelným aktualizacím se Mirror stal oblíbenou volbou pro mnoho vývojářů hledajících efektivní a spolehlivé síťové řešení.

Kromě těchto nástrojů Unity také poskytuje rozhraní pro integraci s externími síťovými službami, jako jsou Firebase, PlayFab a GameSparks, které nabízejí rozšířené možnosti pro backendové služby, databáze, autentizaci uživatelů a analýzu. Tyto služby umožňují vývojářům implementovat komplexní online funkcionalitu, včetně sociálních funkcí, cloudového ukládání a live operací, což zvyšuje možnosti pro tvorbu online her.

V kontextu implementace online her v Unity je zřejmé, že platforma poskytuje bohaté spektrum nástrojů a služeb pro vývojáře. Od základních síťových operací po integraci



s cloudovými službami, Unity umožňuje tvůrcům her efektivně realizovat své vize pro multiplayerové hry. Ačkoliv volba konkrétního nástroje nebo služby závisí na specifických potřebách a cílech projektu, Unity nabízí flexibilní a mocnou platformu pro vytváření online herních zážitků. (Ivashchenko, 2023; Unity Technologies, 2024; Photon, 2024; Mirror Team, 2023)

### 3.6.2 Multiplayerový vývoj v Unreal Engine

Implementace online her v Unreal Engine se významně liší od jiných herních enginů díky své vysoké úrovni integrace síťových funkcí přímo do jádra enginu. Unreal Engine poskytuje robustní nástroje a API pro vývoj multiplayerových her, což umožňuje vývojářům vytvářet škálovatelné, výkonné a realistické online herní zážitky.

Jedním z klíčových prvků Unreal Engine pro vývoj online her je jeho systém replikace. Tento systém je navržen tak, aby automaticky synchronizoval stav hry mezi serverem a připojenými klienty, což zajišťuje, že všichni hráči mají konzistentní zážitek. Replikace v Unreal Engine pokrývá objekty, postavy, animace, a dokonce i fyzikální interakce, což umožňuje vytvářet komplexní a dynamické online světy. Díky robustnímu API mohou vývojáři detailně ovládat, které objekty se mají replikovat a jak často, což je nezbytné pro optimalizaci výkonu a minimalizaci síťového zpoždění.

Architektura online her v Unreal Engine je založena na klient-server modelu, kde server řídí herní logiku a stav hry, zatímco klienti poskytují uživatelské rozhraní a vstupy. Unreal Engine umožňuje snadno definovat rozdílné role pro objekty ve hře (např. Autoritativní Server, Klient) a podporuje koncepty jako jsou predikce na straně klienta a opravy na straně serveru, což je klíčové pro snížení vnímaného lagu v rychlých multiplayerových hrách.

Pro vývojáře, kteří upřednostňují vizuální programování, Unreal Engine nabízí síťové funkce také ve svém systému Blueprint. Toto umožňuje designérům a vývojářům bez hlubokých znalostí C++ vytvářet a testovat síťové herní mechaniky pomocí vizuálního skriptování. Blueprints podporuje vytváření volání vzdálených procedur (RPCs) a manipulaci se stavem replikovaných objektů, což je zásadní pro rychlé prototypování a iteraci multiplayerových funkcí.

Unreal Engine poskytuje rozsáhlou podporu pro vývoj a nasazení dedikovaných serverů, což je základ pro mnoho profesionálních online her. Vývojáři mohou v Unreal

Engine snadno kompilovat a spouštět dedikované servery, což umožňuje efektivní správu herních session, autentizaci hráčů a matchmaking. Tato podpora zahrnuje také nástroje pro monitorování a optimalizaci výkonu serveru, což je klíčové pro udržení stabilního a plynulého online herního prostředí.

Unreal Engine byl navržen s ohledem na multiplatformní hraní, což umožňuje hráčům připojit se k hře z různých zařízení a platforem bez kompromisů v herním zážitku. Podpora pro cross-platform play zahrnuje komplexní síťové testování a optimalizaci, aby se zajistilo, že hra běží hladce na všech podporovaných zařízeních, což rozšiřuje dosah a dostupnost online her vytvořených v Unreal Engine.

Implementace online her v Unreal Engine vyžaduje hluboké porozumění síťovému programování a architektuře, ale díky bohaté sadě nástrojů a podpoře enginu mohou vývojáři vytvářet vysoce kvalitní, škálovatelné a imerzivní multiplayerové zážitky. Tyto nástroje a technologie nejen usnadňují vývoj online her, ale také poskytují základ pro inovace a experimentování s novými formami online interakce a hraní. (Epic Games, 2024)

### **3.6.3 Shrnutí online vývoje v obou platformách**

Unreal Engine je vysoce ceněn pro pokročilé síťové možnosti, které umožňují vývojářům vytvářet detailní a vizuálně působivé online světy s komplexními interakcemi. Jeho podpora pro dedikované servery a robustní architektura klient-server umožňují tvorbu škálovatelných a výkonných online her. Unreal Engine také nabízí silné nástroje pro vizuální skriptování pomocí Blueprint, což usnadňuje prototypování a vývoj multiplayerových funkcí bez hlubokých znalostí programování.

Na druhé straně, Unity poskytuje vývojářům flexibilní a přístupnou platformu pro vývoj online her, s podporou pro různé síťové frameworky jako Photon Unity Networking (PUN) a Mirror, které nabízejí různé úrovně abstrakce a škálovatelnosti. Unity je obzvláště populární mezi nezávislými vývojáři a malými studii, částečně díky jeho snadné integraci s různými cloudovými službami a široké podpoře pro multiplatformní vývoj. Tento engine umožňuje rychlé prototypování a efektivní iteraci, což je klíčové pro dynamický vývoj online her.

Oba enginy nabízejí podporu pro cross-platform play, což umožňuje hráčům na různých zařízeních a platformách užívat si společný herní zážitek. Tato schopnost je stále důležitější v dnešním propojeném světě, kde hráči očekávají bezproblémové online

interakce bez ohledu na platformu. Jak Unreal, tak Unity se neustále vyvíjejí, aby poskytovaly lepší nástroje a funkce pro vývoj online her, reagující na rostoucí požadavky vývojářů a hráčů.

V závěru, výběr mezi Unreal Engine a Unity pro vývoj online her závisí na řadě faktorů, včetně specifických požadavků projektu, preferencí týmu ve vývojových nástrojích, a očekávané úrovni grafické kvality a výkonu. Unreal Engine může být preferován pro projekty, které vyžadují pokročilé grafické schopnosti a komplexní síťové interakce, zatímco Unity může být vhodnější pro vývojáře hledající flexibilitu a rychlou iteraci s širokou multiplatformní podporou. Obě platformy poskytují silný základ pro tvorbu online her a mají silné komunity, které nabízejí podporu a zdroje pro vývojáře.

## 4 Vlastní práce

Nejvíce přímočarým způsobem, jak porovnat tyto dva enginy je provedením zátěžového testování. Cílem toho porovnání je ověřit, který z enginů pracuje lépe s výkonem počítače. Testování touto metodou poskytuje jednoznačné číselné hodnoty, které umožňují je porovnat. Testování bylo provedeno ve třech následujících kategoriích:

1. Výkon fyzikálních výpočtů
2. Výkon grafického renderování
3. Využití operační paměti

Pro jednotlivé testy byla sestrojena testovací aplikace zaměřená na konkrétní problematiku.

Všechny testy byly provedeny na následující hardwarové konfiguraci:

- Procesor: Intel(R) Core(TM) i5-7400 CPU @ 3.00GHz
- Paměť: 16,0 GB; rychlost: 2400 MHz
- GPU: NVIDIA GeForce GTX 1060 6 GB
- Operační systém: Windows 10 64bit
- Obrazovka: šířka 1920 pixelů, výška 1080 pixelů, frekvence 144 Hz

Pro sledování využití systémových prostředků byl použit software MSI Afterburner (Micro-Star INT'L). Tento software je volně dostupný a nabízí široké možnosti monitorování výkonu hardware.

### 4.1 Verze enginů

Pro veškeré testy bylo použito Unity verze 2022.3.20. Ta je v době psaní práce nejnovější verzí s dlouhodobou podporou. Je tedy nejvhodnější ji používat pro větší projekty. Pro testování aplikace vytvořené v Unreal Engine byla použita nejnovější verze 5.3.

### 4.2 Výkon fyzikálních výpočtů

V této kapitole je ukázáno, jak jsou enginy výkonné v situaci náročné pro výpočet fyzikálních kolizí. Pro porovnání byl měřen výkon procesoru, počet renderovaných snímků

za sekundu a využití operační paměti. Většina fyzikálních výpočtů je prováděna pomocí procesoru. V testech je tedy očekáváno, že docházelo k jeho vysoké zátěži. Fyzikální výpočty hrají klíčovou roli v mnoha hrách, a proto je důležité, aby je herní engine prováděl efektivně.

#### **4.2.1 Metodika testování**

V obou herních enginech byla vytvořena identická aplikace. V této aplikaci jsou postupně vytvářeny pohybující se kostky. Všechny kostky se pohybují směrem k jednomu bodu a vzájemně kolidují. Se zvětšujícím se počtem kostek narůstá také počet kolizí a zátěž fyzikálních výpočtů. Sledována byla zátěž procesoru, využití paměti a počet renderovaných snímků za sekundu.

#### **4.2.2 Postup při průběhu testu**

- 1) Zkontrolujeme stálou teplotu v místnosti a vypneme všechny ostatní nepotřebné procesy na zařízení.
- 2) Spuštění aplikace.
- 3) Poté co je scéna načtena a objekty vytvořeny, se čeká několik minut. V této době se stabilizuje počet kolizí.
- 4) Pomocí měřicího software je zaznamenán výsledek po dobu deseti minut.

Po provedení jednotlivých měření byl počítač chvíli nevyužíván, aby klesla teplota hardwaru na stejnou počáteční hodnotu. Také byly vypnuty veškeré ostatní procesy, které by mohly ovlivnit výsledek.

#### **4.2.3 Unity**

V Unity byl vytvořen prázdný projekt využívající renderovací systém URP (Universal Render Pipeline), který je určen pro méně náročné úlohy a vyžaduje menší výkon. Byl zvolen pro menší dopad na výsledky test, než které by měl HDRP (High definition render pipeline).

Po spuštění aplikace jsou postupně v náhodných pozicích vygenerovány kostky (viz obrázek 10). Jednotlivé kostky byly vytvořeny jako „Prefab“. To je koncept, který umožňuje

vytvořit objekty mimo konkrétní scénu. Takto vytvořené objekty, lze snadno opakovaně použít. Představují jakýsi vzor, podle kterého jsou vytvářeny jeho klony.

Každá z těchto kostek je navázána na skript, který v pravidelných intervalech nastavuje rychlost na určitou hodnotu a směr rychlosti pohybu do zvoleného středu. Tímto je pohyb nahodilý, ale zároveň se kostky pohybují vždy v blízkosti, a proto dochází k velkému množství kolizí.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

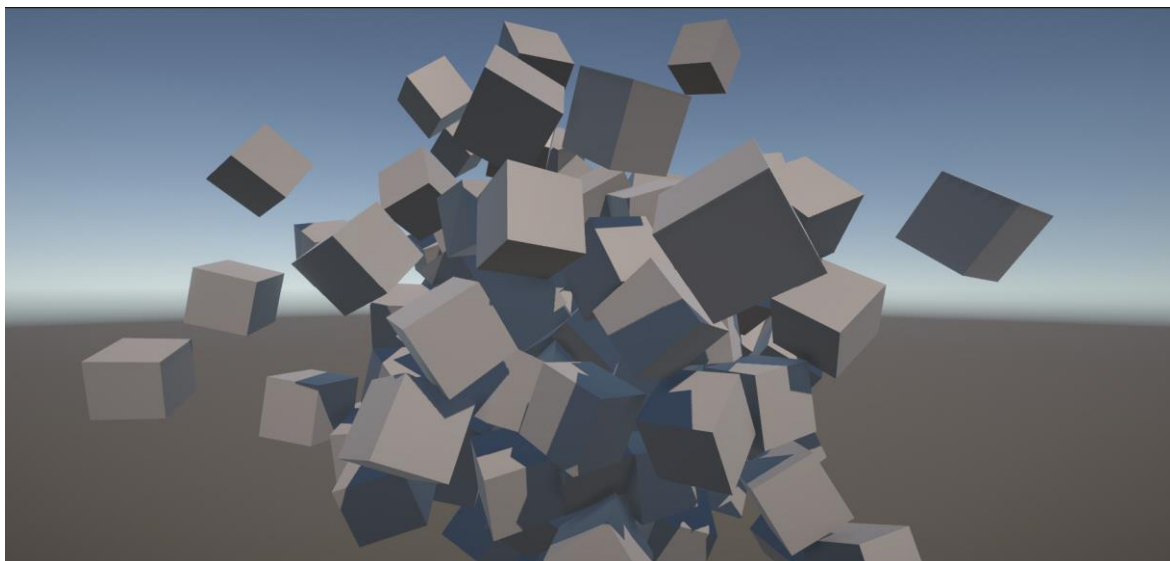
public class Cube : MonoBehaviour
{
    Rigidbody rb;
    const float speed = 20f;
    const float frequency = 2f;

    void Awake()
    {
        rb = GetComponent<Rigidbody>();
        InvokeRepeating("MoveTowards", 0, frequency);
    }

    void MoveTowards()
    {
        rb.velocity = -transform.position.normalized * speed;
    }
}
```

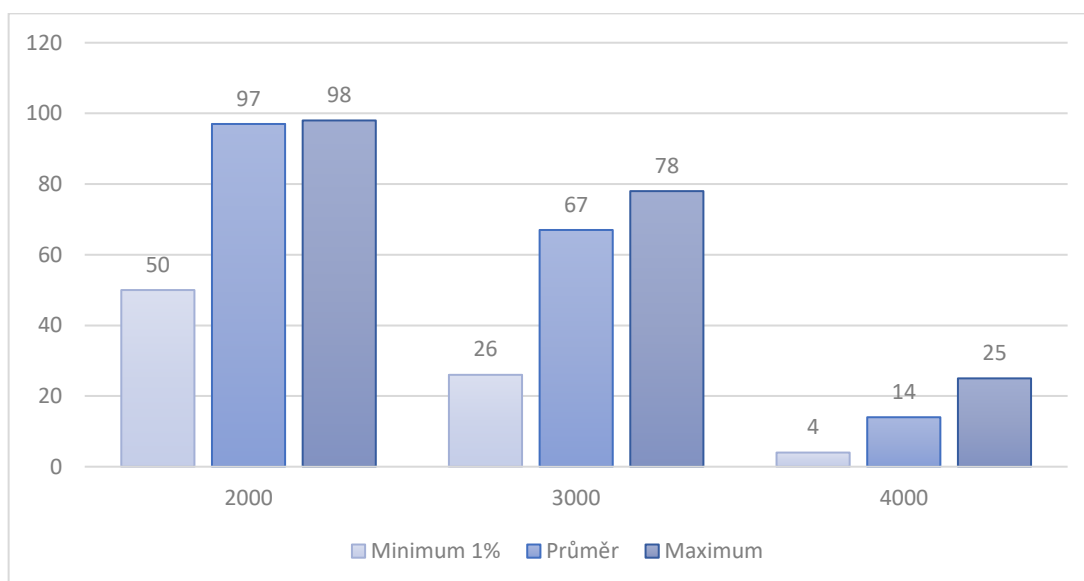
*Obrázek 9 Skript pohybující s testovacími objekty*

Po spuštění programu bylo provedeno několik měření. Měřena byl počet zobrazovaných snímků za sekundu, zátěž procesoru a využití paměti. Při měření bylo dosaženo očekávaných výsledků. Větší množství kolizí mělo za dopad dosažení menšího počtu snímků za sekundu a větší zátěže procesoru.



Obrázek 10 Vizualizace kolidujících objektů

Pro zvýšení přesnosti testu zaměřeného na náročnost výpočtu fyzikálních kolizí nebyly kostky v průběhu testu renderovány. Tím bylo zajištěno využití větší části výpočetního výkonu na zkoumanou problematiku.



Graf 1 Počet snímků za sekundu v provedených testech v Unity

Graf 1 obsahuje naměřená data z testů. Z naměřených snímků za sekundu stojí za zmínku, to že se hodnota počtu průměrných snímků za sekundu pro nižší množství kolizí pohybovala blízko maxima. Při větším množství kolizí docházelo k více nestabilnímu počtu snímků za sekundu.

Tabulka 2 Výsledky ostatních měření v testu Unity

	<b>Zátěž procesoru</b>	<b>Zátěž grafické karty</b>	<b>Využití paměti</b>
<b>2000</b>	53%	33%	5551 MB
<b>3000</b>	68%	31%	5868 MB
<b>4000</b>	68%	27%	5974 MB

Z výsledků měření využití hardwarových zdrojů, vidíme mírně stoupající využití procesoru. Využití grafické karty se pohybuje stále kolem stejné hodnoty. Nejjednoznačnější nárůst lze vidět ve využití operační paměti. Ten můžeme vysvětlit nutností uložit větší množství aktivních objektů v aplikaci.

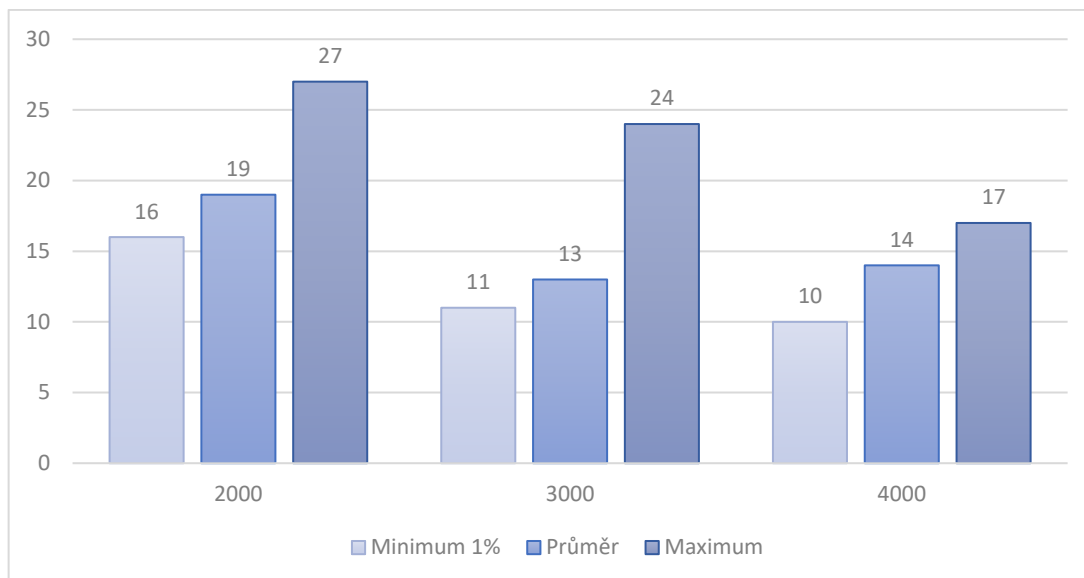
Dosažené výsledky jsou závislé na testovacím hardware. Proto nebylo možné testovat extrémnější případy s vysokým množstvím kolizí. Tato simulace si ale klade za cíl porovnat vlastnosti herního engine v situacích v kterých bude často používán. Tedy náročnosti simulace lze připodobnit k náročnosti hry využívající fyzikální výpočty. S tímto můžeme prohlásit simulaci za dostatečnou.

#### 4.2.4 Unreal Engine

V Unreal Engine byl vytvořen prázdný projekt využívající C++ pro lepší optimalizaci výpočtů. Byla vytvořena co nejvěrnější kopie aplikace testované v Unity. Proces tvorby aplikace v Unreal Engine je velice odlišný od Unity, stejně tak odlišný je přístup optimalizaci a výkonu. Unreal Engine například preferuje vizuální kvalitu, a to očekávaně vede k větší náročnosti pro výpočetní výkon. S tímto předpokladem lze očekávat rozdílné výsledky i v testu fyzikálních výpočtů. Jelikož při průběhu simulace nedochází jen k výpočtu kolizí ale také k veškerým dalším výpočtům, které engine sám o sobě provádí.

Obdobě jako bylo popsáno v předchozí kapitole o testu v Unity, byl v Unreal Engine vytvořen skript pro pohyb kostek.





Graf 2 Počet snímků za sekundu v provedených testech v UE

Stejně jako v Unity byly nebyly v průběhu simulace kostky renderování pro větší přesnost měření výkonu fyzikálních výpočtů. Z naměřených výsledků je vidět že počet snímků za sekundu je nízký i při menším počtu kolizí. U většího počtu kolizí bylo naměřeno stabilní počet snímků. Důležité je zmínit že na rozdíl od Unity při delším běhu simulace, než bylo testováno doházelo k následnému postupnému snižování počtu snímků za sekundu.

Tabulka 3 Zátěž hardware v průběhu testu v Unreal Engine

	Zátěž procesoru	Zátěž grafické karty	Využití paměti
2000	46 %	32 %	5011 MB
3000	45 %	34 %	5343 MB
4000	43 %	30 %	5417 MB

Z výsledků změřené zátěže hardware můžeme pozorovat stabilní zátěž procesoru i grafické karty. Nižší zatížení procesoru lze vysvětlit tím že Unreal Engine využívá k fyzikální výpočtů pouze jedno procesorové jádro, a proto není schopen v této konkrétní simulaci efektivně využít celý potenciál výpočetní síly hardware.

### 4.3 Výkon grafického renderování

V následujících podkapitolách jsou popsány testy porovnávající herních enginů Unity a Unreal Engine při vykreslování modelů. Cílem je porovnat rozdíl mezi využitím hardware

při stejném testu. V obou enginech byly vytvořena scéna se stejnými modely pro co nejpřesnější porovnání. Nastavení grafiky byla ponechána na výchozích hodnotách.

Všechny testy jsou provedeny na následující hardwarové konfiguraci:

- Procesor: Intel(R) Core(TM) i5-7400 CPU @ 3.00GHz
- Paměť: 16,0 GB; rychlost: 2400 MHz
- GPU: NVIDIA GeForce GTX 1060 6 GB
- Operační systém: Windows 10 64bit
- Obrazovka: šířka 1920 pixelů, výška 1080 pixelů, frekvence 144 Hz

Pro sledování využití systémových prostředků je použit software MSI Afterburner (Micro-Star INT'L). Tento software volně dostupný a nabízí široké možnosti monitorování výkonu hardware.

#### **4.3.1 Metodika testování**

V obou herních enginech byla vytvořena identická aplikace. Pro test statických modelů byla vytvořena scéna s detailními modely statické vegetace. Pro test dynamického renderování byla vytvořena scéna s animovanými postavami s cyklicky opakující se animací. Sledována bude zátěž procesoru, využití paměti a počet vykreslených snímků za sekundu. Pro každý test bylo provedeno 5 měření a výsledky byly zprůměrovány.

#### **4.3.2 Postup při průběhu testu**

- 1) Zkontrolujeme stálou teplotu v místnosti a vypneme všechny ostatní nepotřebné procesy na zařízení.
- 2) Spuštění aplikace.
- 3) Poté co je scéna načtena a objekty vytvořeny, se čeká několik minut. Než se vyrovná teplota zařízení.
- 4) Pomocí měřicího software je zaznamenán výsledek po dobu deseti minut.

Po provedení jednotlivých měření byl počítač chvíli nevyužíván, aby klesla teplota hardwaru na stejnou počáteční hodnotu. Také byly vypnuty veškeré ostatní procesy, které by mohly ovlivnit výsledek.

### 4.3.3 Výsledky testu vykreslování statických modelů

V obou enginech byly připraveny identické scény se statickou scénérii. Následně byly sestaveny aplikace s těmito scénami. Aplikace byly spuštěny za stejných podmínek na stejném hardware.

#### 4.3.3.1 Unity

Do Unity byly naimportovány modely volně dostupné ke stažení z ukázkového projektu Unity HDRP (Dostupné ze stránek Unity). Pro renderování byl použit nastavení renderování High Definition Render Pipeline (HDRP). Toto nastavení zajišťuje největší kvalitu obrazu a je určeno především pro aplikace vytvářené pro počítače.

Při sestavování scény byl vyžit princip Prefab, který v unity slouží k ukládání vytvořených objektů na disk mimo aktuální scénu. Takto vytvořené objekty lze skupinově upravovat a znovu použít.

Tabulka 4 Využití hardwarových zdrojů ve statickém testu v Unity

Grafická zátěž	Zátěž procesoru	Využití operační paměti
87 %	41 %	6163 MB

Z naměřených hodnot využití hardwarových prostředků vidíme zvýšené využití grafického jádra. To je očekávané vzhledem k testovanému problému. Hodnota využití procesoru se pohybovala pod 50 %. Využití operační paměti výrazně přesáhlo hodnoty měřené v předchozím testu kolizí, a to lze vysvětlit nutností práce s větším množstvím textur.

Tabulka 5 Naměřené hodnoty snímků za sekundu ve statickém testu v Unity

Průměrné snímky za sekundu	Maximální snímky za sekundu	1 % nejnižších snímků za sekundu
96	100	37

Počet průměrně vykreslených snímků za sekundu dosahoval devadesáti šesti. Tato hodnota se blížila maximální naměřené hodnotě sto snímků za sekundu. Z toho můžeme

odvodit, že počet snímků byl v průběhu testu relativně stabilní ohledně maxim. To je očekávané u statického testu, kde nedochází k větším změnám.

Jedno procento nejnižších vykreslovaných snímků bylo pod nízkou hodnotou třicet sedm snímků za sekundu. To znamená že jedno procento času z měřených testů byly snímky za sekundu pod touto hodnotou. Unity využívá ke správě paměti mechanismus garbage collection, který pravidelně spouštěn pro uvolnění nevyužívané paměti. Tomu lze připsat tyto minima, jelikož při spuštění tohoto správce paměti výrazně stoupne výpočetní čas snímků, a tedy i klesne počet vykreslovaných snímků za sekundu.

#### 4.3.3.2 Unreal Engine

V Unreal engine byla sestrojena identická scéna té, která byla využita pro testování v Unity. Byly naimportovány modely využitě v testu Unity a sestavena aplikace pro počítač.

Pro práci s objekty byl využit princip Blueprint, který má obdobnou funkcionalitu jako Prefab v Unity. Liší se především rozpoštěním editoru a postupem práce s ním. Dalším rozdílem je možnost vizuálního skriptování, která byla využita pro práci s materiály a jejich nastavení.

*Tabulka 6 Využití hardwarových prostředků ve statickém testu v Unreal Engine*

<b>Grafická zátěž</b>	<b>Zátěž procesoru</b>	<b>Využití operační paměti</b>
98 %	58 %	5843 MB

Z výsledků využití hardware při statickém testu v Unreal Engine je vidět že limitujícím faktorem byla grafická karta. Unreal engine byl velice náročný pro grafické výpočty. Zátěž procesoru se také pohybovala ve větších hodnotách, ale nepředstavovala limitující faktor pro počet vykreslovaných snímků za sekundu. Ovšem velkým rozdílem od Unity je nižší náročnost na operační paměť.

*Tabulka 7 Využití hardware při testu statického renderování v Unreal Engine*

<b>Průměrné snímky za sekundu</b>	<b>Maximální snímky za sekundu</b>	<b>1 % nejnižších snímků za sekundu</b>
71	78	39

Počet naměřených vykreslených snímků za sekundu se v testu Unreal Engine pohybovat kolem nižších hodnot než v předchozím testu Unity. Tomu odpovídá také větší využití grafické karty, jejíž výkon byl limitujícím faktorem. Průměr se pohyboval blízko maxima, to bylo očekávané u statického testu. Při nejnižší jedno procento naměřených hodnot se pohybovalo pod třiceti devíti snímky za sekundu.

#### 4.3.4 Výsledky testu vykreslování dynamických modelů

V Unity a Unreal Engine byla sestrojena identická aplikace s osmdesáti animovanými postavami, viz. obrázek 11.



Obrázek 11 Ukázka scény s animovanými postavami

##### 4.3.4.1 Unity

Byla sestrojena scéna s animovanými postavami. Využity byly volně dostupné postavy od společnosti Reallusion, které jsou volně ke stažení na jejich webových stránkách.

Tabulka 8 Využití hardwarových zdrojů ve dynamickém testu v Unity

Grafická zátěž	Zátěž procesoru	Využití operační paměti
98 %	40 %	7096 MB

V tabulce 8 vidíme využití hardwarových prostředků. Docházelo k vysokému využití grafické karty. Animované postavy byly výrazně náročnější při vykreslování než statické objekty. Hodnota využití procesoru se pohybovala kolem 40 %. Množství využití operační paměti přesáhlo všechny dosud naměřené hodnoty a lze vysvětlit množstvím renderovaných postav.

Tabulka 9 Naměřené hodnoty snímků za sekundu ve dynamickém testu v Unity

<b>Průměrné snímky za sekundu</b>	<b>Maximální snímky za sekundu</b>	<b>1 % nejnižších snímků za sekundu</b>
87	90	39

Tabulka 9 obsahuje počet průměrně vykreslených snímků za sekundu. Ten se pohyboval okolo osmdesáti sedmi. Tato hodnota se blížila maximální naměřené hodnotě devadesát snímků za sekundu. Limitujícím faktorem snímků bylo využití grafické karty.

Jedno procento nejnižších vykreslovaných snímků se bylo pod nízkou hodnotou třicet devět snímků za sekundu.

#### 4.3.4.2 Unreal Engine

V Unreal engine byla sestrojena identická scéna. Při vytváření byl využit princip Blueprint pro připravení modelů. Využití animace a modely jsou volně dostupné na internetu.

Tabulka 10 Využití hardwarových zdrojů ve dynamickém testu v Unreal Engine

<b>Grafická zátěž</b>	<b>Zátěž procesoru</b>	<b>Využití operační paměti</b>
98 %	59 %	7271 MB

Z naměřených hodnot využití hardwarových prostředků vidíme vysoké využití grafické karty. Dynamický test byl náročnější při vykreslování než statické objekty. Unreal Engine byl náročnější z pohledu využití procesoru Vyšší množství využití operační paměti lze vysvětlit množstvím renderovaných postav. V porovnání s Unity lze vidět zvýšené nároky.

Tabulka 11 Naměřené hodnoty snímků za sekundu ve dynamickém testu v Unreal Engine

<b>Průměrné snímky za sekundu</b>	<b>Maximální snímky za sekundu</b>	<b>1 % nejnižších snímků za sekundu</b>
64	92	57

Počet průměrně vykreslených snímků za sekundu dosahoval pouze šedesáti čtyř. Hodnota snímků výrazně kolísala podle fází animace. Jedno procento nejnižších

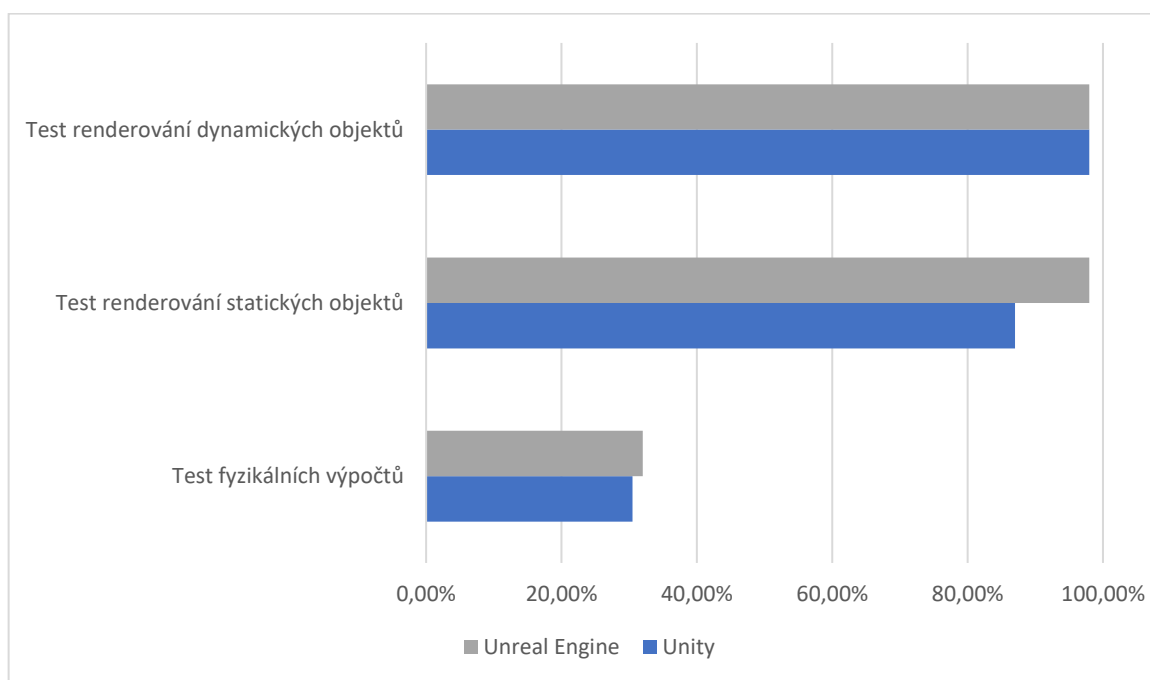
vykreslovaných snímků se pohybovalo relativně blízko průměrné hodnoty pod padesáti sedmi snímky za sekundu.

## 5 Výsledky a diskuse

V této kapitole jsou představeny výsledky studie zdrojů a provedených testů. Výsledky jsou následně interpretovány a je diskutováno nad jejich limity. Následně jsou uvedeny doporučení pro uživatele uvažující o výběru jednoho z těchto herních engineů plynoucí z výsledků.

### 5.1 Výsledky porovnání výkonu herních engineů Unreal Engine a Unity

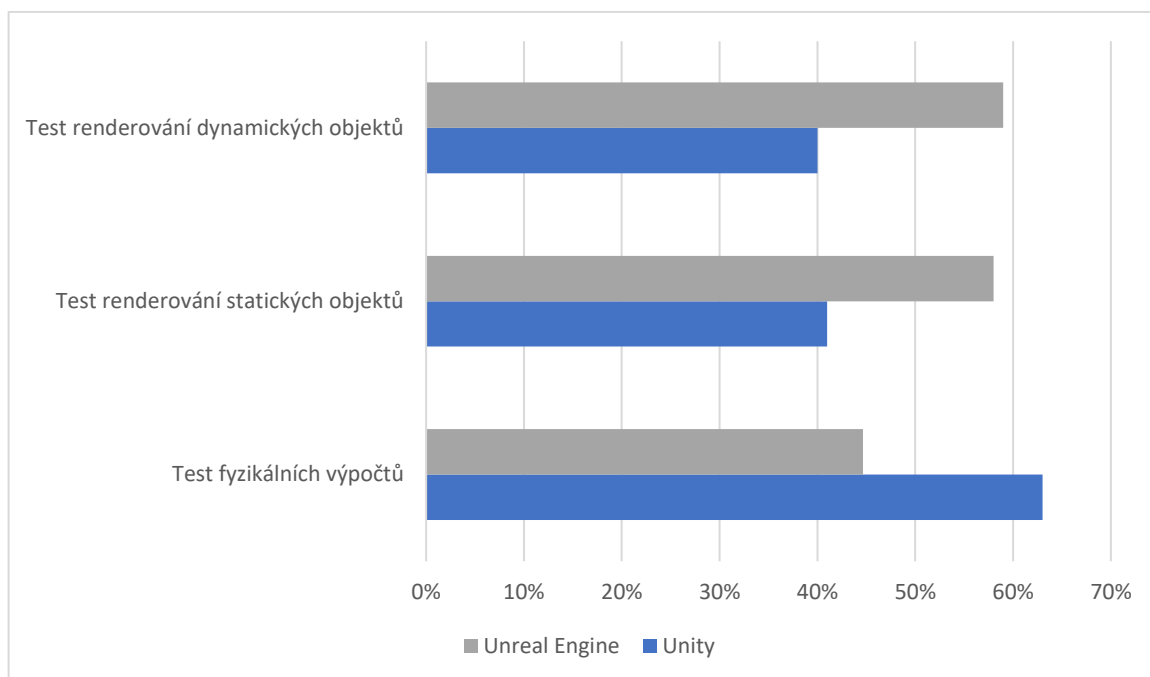
Byly provedeny testy využití hardwaru a měření výkonu identických aplikací sestavených v obou herních enginech. Cílem těchto testů bylo porovnat měřitelným způsobem určité vlastnosti herních engineů Unreal Engine a Unity. U výsledků těchto testů nejsou důležité konkrétní hodnoty v samostatných testech, ale jejich porovnání s hodnotami dosaženými mezi porovnávanými enginey.



Graf 3 Využití grafické karty v jednotlivých testech

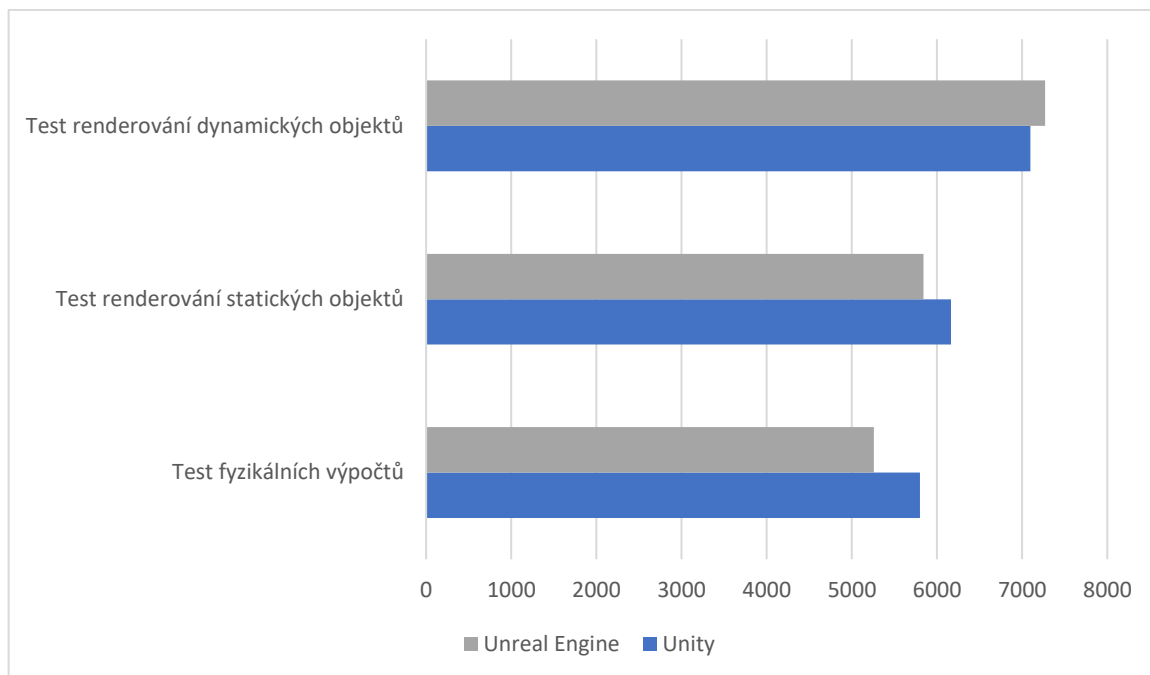
V testu renderování dynamických objektů oba enginey ukazují podobnou hodnotu maximálního využití grafické karty. U ostatních testů lze vidět mírně vyšší nároky u Unreal Engine oproti Unity.





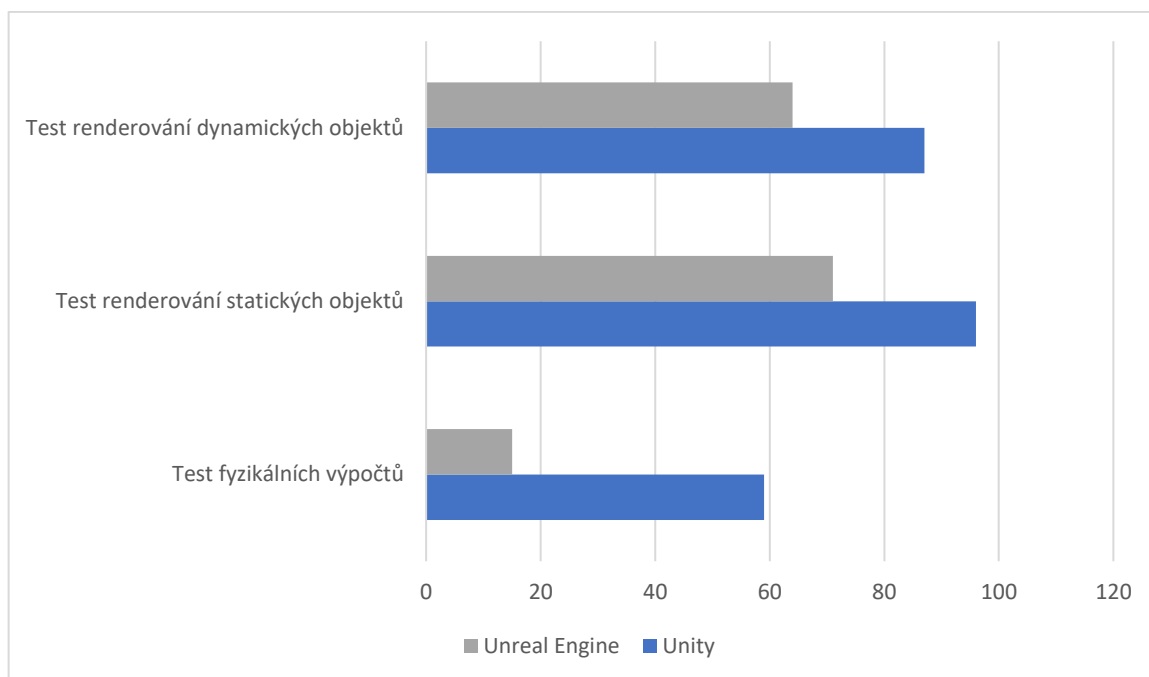
*Graf 4 Využití procesoru v jednotlivých testech*

Graf 4 porovnává náročnost na procesor při provádění různých testů v herních enginech Unreal Engine a Unity. V testu fyzikálních výpočtů je vidět, že Unity je mnohem náročnější na procesorový výkon než Unreal Engine. Zatímco v testech renderování statických i dynamických objektů je náročnost obou enginů poměrně podobná, s tím, že Unity je méně náročné. Tyto výsledky naznačují, že pro projekty, které vyžadují intenzivní fyzikální simulace, může být Unreal Engine vhodnější, avšak jeho použití by mohlo být spojeno s vyššími požadavky na procesor. Na druhou stranu pro projekty s menším důrazem na fyzikální simulace by Unity mohlo být efektivnější a méně náročné na systémové zdroje.



*Graf 5 Využití systémové paměti v průběhu testů*

Na grafu 5 je zobrazeno využití operační paměti během testů Unreal Engine a Unity. V testu fyzikálních výpočtů a renderování dynamických objektů využívá Unity výrazně více paměti než Unreal Engine. U statických objektů je rozdíl v paměťové náročnosti obou enginů vyšší, přičemž Unreal Engine opět vykazuje nižší využití paměti. Výsledky naznačují, že Unreal Engine může být vhodnější pro projekty, kde je kladen důraz na detailní grafiku a pokročilé fyzikální efekty, a při tom jsou omezenější paměťové zdroje. Zatímco u Unity můžeme očekávat větší paměťovou náročnost.



Graf 6 Průměrný počet snímků za sekundu v průběhu testů

Graf 6 ukazuje průměrný počet snímků za sekundu (FPS) během testů v Unreal Engine a Unity. Vyšší hodnota FPS značí lepší plynulost a reaktivitu hry. Unity překonává Unreal Engine ve všech kategoriích testů, s největším rozdílem v testu fyzikálních výpočtů. I při testu renderování statických a dynamických objektů je Unity na lepší úrovni, což naznačuje, že jeho optimalizace pro vykreslování je efektivnější. Výsledky poukazují na to, že Unity může nabídnout hladší herní zážitek, což je důležité pro vývojáře při výběru enginu, zvláště pro akční nebo graficky náročné hry.

Kromě výsledků naměřených využití hardwaru komentáře efektivity práce herních enginů s ním. Je důležité okomentovat vizuální aspekt výsledné aplikace. Přesto že obou enginech byly sestaveny co nejvíce identické aplikace, na první pohled bylo poznat rozdíl mezi kvalitou vykreslení modelu a osvětlení. Unreal Engine je na rozdíl od Unity primárně určen pro zobrazování co nejrealističtějších modelů a je pro to optimalizován. Toto nelze změřit žádnou metrikou ale renderování modely v byly jednoznačně kvalitněji vykresleny a realističtěji nasvětleny. V testech byla použita výchozí nastavení a po větších úpravách lze dosáhnout realističtějších výsledků i v Unity. Měření ovšem byl výsledek „out of the box“, tedy základních herních enginů bez úprav a pluginů.

Na druhou stranu je nutné podotknout že se jedná o velice rozsáhle a komplexní programy se spoustou možností. V praxi lze očekávat že výsledky budou záviset nejen na samotném software ale také na zkušenostech a znalostech jejich uživatele.

## 5.2 Limity výsledků

Před představením závěru srovnání herních enginů Unity a Unreal Engine je důležité se zamyslet nad limity toho srovnání. Jak bylo představeno v předchozích částech práce, lze tyto herní enginy používat při vývoji různorodého množství aplikací. Tato práce se zabývá především herními aplikacemi, ale v praxi jsou používány i v jiných profesích jako architektura nebo filmový průmysl. V samotném herním průmyslu pak slouží k vývoji jak relativně jednoduchých dvouprostorových her, tak k velice realistickým simulacím s rozsáhlým detailně propracovaným virtuálním světem. Velká šíře možností pokládá určité limity na závěr a doporučení, která z něj mohou plynout. Každý jednotlivý případ vývoje aplikace bude mít specifické nároky. Z tohoto důvodu nelze dojít k jednoznačně obecnému závěru.

## 5.3 Interpretace výsledků

Hlavním cílem testování výkonu herních enginů bylo zjištění efektivita herních enginů z pohledu práce s hardwarem. Porovnávány byly dva největší herní enginy z pohledu podílu na trhu. Z naměřených výsledků vyplynulo, že aplikace vytvořené v Unreal Engine využívaly více výpočetního výkonu procesoru a dosahovaly nižšího počtu snímků za sekundu. Unity dosahovalo výrazně vyššího počtu snímků za sekundu. Tedy výsledkem měření je, že aplikace vytvořené Unity dosahují většího počtu snímků za sekundu a využívají menší množství výpočetního výkonu procesoru.

Na základě provedeného výzkumu a získaných výsledků lze konstatovat, že oba enginy, Unreal Engine i Unity, umožňují vytvářet hry se solidní grafikou a výkonem. Ve srovnání s Unity však Unreal Engine poskytuje snazší přístup k modelům a nástrojům pro tvorbu her s vysokou kvalitou vizuálních efektů na úkor výkonu. Na druhou stranu Unity umožňuje vytvářet hry s podobnými funkcemi a vyšším výkonem, ale horšími vizuálními efekty. Je však třeba vzít v úvahu potenciální vliv zvyšující se složitosti hry na úpravy herního výkonu, což by mohlo vést k posunu výsledků ve prospěch druhé strany.

## 5.4 Finanční srovnání

Jednou z důležitých stránek při srovnání herních enginů z pohledu herního studia jsou finanční náklady spojené s používáním toho herního enginu. V kapitolách 3.3.2 a 3.4.2 byly představeny modely financování porovnávaných herních enginů. Oba herní enginy nabízejí několik variant licence použití. Oba také nabízejí možnost vývoje a publikování aplikací s licenci bez poplatků. U Unity se tato licence nazývá Personal a u Unreal Standard. Unity omezuje využití této licence pro společnosti s příjmem do dvou set tisíc dolarů za kalendářní rok. U Unreal Engine je licence Standard omezena také výší příjmu, ale tento příjem je počítán pouze z konkrétní aplikace. Výše tohoto limitu je jeden milión dolarů za celou existenci aplikace. Z pohledu menších projektů a společností bude v mnoha případech benevolentní limit jednoho miliónu dolarů představovat jednoznačnou výhodu.

Přístup k placeným licencím je výrazně odlišný u obou programů. Unity zvolilo přístup pravidelných poplatků za jednotlivé uživatele, a proto bude finančně nákladnější pro větší společnosti. Ceny licencí se pohybují v jednotkách tisíc dolarů za osobu za rok. U projektů přesahujících limit jednoho miliónu dolaru a zároveň jednoho miliónu stažení je také placen poplatek 2,5 % z příjmů z aplikace. Unreal Engine nabízí jednodušší variantu. Poplatky jsou placené z příjmu aplikace, pokud dosáhne za dobu své existence hodnotu jednoho miliónu dolarů. Poplatek činí 5 %. Tento poplatek není omezen počtem stažení jako je tomu u Unity. Za předpokladu, že oba herní enginy nabízejí srovnatelné produkty v placené verzi, můžeme konstatovat, že pro menší společnosti bude využití Unity levnější díky nižším poplatkům.

## 5.5 Shrnutí Unity

Pro začátečníky bude pravděpodobně nejlepší volbou engine Unity. Velmi dobře napsaná dokumentace, mnoho kurzů a připravených šablon vám umožní se seznámit se všemi nejčastějšími důležitými funkcemi enginu poměrně rychle. Kromě toho nabízí podporu největšího množství platforem a velká univerzálnost enginu vám umožní realizovat hry jakéhokoli žánru. Unity je jediným enginem se specializovaným 2D režimem, takže bude nejlepší volbou pro uživatele, kteří chtějí vytvořit dvourozměrnou hru. Nevýhodou tohoto enginu je bezpochyby relativně malý počet specializovaných editorů. Moduly umožňující animaci, tvorbu video scén, nebo přidávání zvukových efektů ve srovnání s konkurencí umožňují pouze základní operace. Engine lze rozšířit pomocí zásuvných modulů a knihoven, ale mnohé z nich jsou placené, což může potenciální uživatele odradit. Co se týče kvality grafiky, zdá se, že engine Unity je v tomto slabší.

## 5.6 Shrnutí Unreal Engine

Unreal Engine narozdíl od Unity nemusí být vhodnou volbou pro začátečníky. Veliká šíře dostupných možností v editoru může být zahlcující pro nové uživatele při prvním kontaktu s tímto nástrojem. Unreal Engine může být ale dobrou volbou pro uživatele bez programátorských schopností. Nástroj BluePrints umožňuje sepsání celé logiky hry pomocí grafických diagramů, které mohou kompletně nahradit nutnost programování v jazyce C++. Unreal Engine se od konkurence odlišuje podporou mnoha cílových hardwarových platforem, schopností generovat vysoce kvalitní grafiku, osvědčeným a poměrně snadno použitelným síťovým modulem, skvělým editorem materiálů, možností vytvářet pokročilé skeletové animace a systémy podporujícími implementaci umělé inteligence.

## 5.7 Doporučení

V této kapitole budou doporučeny případy, pro které je vhodné využít jednotlivé srovnávané herní enginy, Pro ilustraci konkrétních případů budou představeny tři fiktivní zájemci o využití herních enginů. Každý z nich má konkrétní představu o aplikaci, kterou budou vyvíjet.

Malá začínající firma chce vyvinout mobilní hru. Tato hra bude využívat dvouprostorovou grafiku. Hráč se ve hře stará o svoji vesnici, staví budovy a snaží se dosáhnout co nejvyššího skóre. Pro hru tohoto typu není prioritou kvalitní 3D grafika ani není podstatné dosahovat vysokého počtu snímků za sekundu. Na základě studie enginů bylo zjištěno, že Unity nabízí na rozdíl od Unreal Engine dedikovaný dvouprostorový mód pro vývoj her. Dále je orientováno na stroje s menším výpočetním výkonem. I z pohledu financí lze doporučit využití herního engine Unity.

Další společností je mezinárodní společnost s několika sty zaměstnanci. Tato společnost hledá herní engine pro vývoj jejich nového herního titulu. Tato hra bude využívat nejmodernější technologie. Hra se bude odehrávat v realisticky ztvárněném virtuálním světě, kde lidé budou moci žít druhý život. Hra bude spustitelná přes brýle pro virtuální realitu. K dispozici bude na brýlích Meta Quest, Playstation VR a HTC Vive. Pro tuto společnost lze doporučit využití herního engine Unreal Engine. Pro práci s realistickou grafikou a nasvětlením je lepší volbou.

Pro třetí ilustrační případ bude uvedena střední společnost, s cílem vytvořit kvalitní vojenský simulátor z pohledu první osoby. V tomto případě lze také doporučit Unreal Engine, jelikož je historicky koncipován primárně pro hry z pohledu první osoby a pro střílečky. Simulátor také nabude větší uvěřitelnosti díky realistickému osvětlení, stínům a grafickým modelům.

Obecně lze říci že hlavními faktory při volbě herního engine jsou: očekávané grafické provedení, finanční dispozice společnosti a zaměření na konkrétní platformy. Unreal Engine lze doporučit pro projekty s více realistickou grafikou pro výkonnější platformy jako osobní počítač, konzole Xbox a PlayStation. Ideální je také pro nulové počáteční náklady, nezávisle na velikosti společnosti. Unity je naopak preferovanou volbou pro projekty na méně výkonné platformy jako mobilní telefon. Je jednoznačnou volbou pro dvoudimenzionální aplikace.

## **5.8 Přínos**

Cílem této práce bylo vytvořit přehled a srovnání dvou herních enginů Unity a Unreal Engine. Tato srovnání si kladla za cíl umožnit snazší rozhodování při volbě herního engine pro nadcházející projekt. Přidanou hodnotou pro čtenáře je ucelené sjednocení všech základních informací relevantních pro rozhodování do jedné práce. Na základě přečtení této

práce by měl čtenář být schopen informovaně zvolit herní engine vhodný pro jeho projekt při vývoji herní aplikace.



## 6 Závěr

Tato diplomová práce představila podrobné srovnání dvou předních herních enginů Unity a Unreal Engine. Hlavním cílem této práce bylo vytvořit srovnání těchto herní enginů za účelem usnadnění rozhodování mezi využitím těchto dvou herních enginů. Zvoleny byly herní enginy Unity a Unreal Engine pro jejich dominantní postavení na herním trhu.

První část práce byla zaměřena na teoretická východiska, která zahrnovala úvod do herního vývoje, základní komponenty herních enginů a jejich představení. Další kapitoly se věnovaly detailnímu představení srovnávaných herních enginů Unity a Unreal Engine včetně jejich historie, cenové politiky, dostupnosti návodů a assetů, a podpory tvorby her pro různé platformy. V neposlední řadě byly představeny možnosti vývoje online her.

V praktické části byly sestaveny prototypy her za účelem porovnání výkonu srovnávaných herních enginů. Porovnány byly z pohledu náročnosti fyzikálních výpočtů a náročnosti grafického renderování. Následně byly představeny výsledky.

Z analýzy a porovnání vyplývá, že oba enginy mají své jedinečné výhody a nevýhody a výběr mezi nimi by měl záviset na specifických požadavcích a cílech projektu. Unity se ukázalo být výbornou volbou pro projekty s omezenějším rozpočtem a pro vývojáře hledající flexibilitu a rychlost vývoje díky rozsáhlé podpoře a dostupnosti návodů a assetů. Na druhou stranu Unreal Engine poskytuje pokročilé grafické a renderovací schopnosti, které jsou ideální pro vytváření vizuálně náročných her a aplikací.

Oba herní enginy nabízejí rozsáhlou podporu pro multiplatformní vývoj, což umožňuje vývojářům dosáhnout širokého spektra hráčů na různých zařízeních. Výběr mezi Unity a Unreal Engine by tedy měl být založen na konkrétních potřebách projektu, přičemž je třeba zvážit faktory jako jsou grafické požadavky, rozpočet, předchozí zkušenosti týmu s danými enginy a cílové platformy.

Vzhledem k rychlému vývoji technologií a průmyslu herního vývoje je také důležité sledovat aktuální trendy a aktualizace těchto enginů, které mohou přinést nové možnosti a optimalizace pro tvorbu her.

Na závěr je možné konstatovat, že ať už se společnost rozhodne pro Unity nebo Unreal Engine, klíčem k úspěchu je hluboké porozumění nástrojům a technikám, které každý engine nabízí, a schopnost využít je k maximálnímu potenciálu pro realizaci herního projektu.

## 7 Seznam použitých zdrojů

- 1 ACADEMIC ACCELERATOR, 2024. Unity Game Engine. In: *Academic Accelerator* [online]. [cit. 2024-01-24]. Dostupné z: <https://academic-accelerator.com/encyclopedia/unity-game-engine>
- 2 ANDRÝSEK, Richard, 2019. *TOP 5 Herních enginů pro indie vývojáře* [online]. [cit. 2024-03-05]. Dostupné z: <https://gamesdev.cz/top-5-hernich-enginu-pro-indie-vyvojare/>
- 3 APPERSON, Lem, 2023. Beginning Game Development: AI Navigation. In: *Medium* [online]. [cit. 2024-02-06]. Dostupné z: <https://medium.com/@lemapp09/beginning-game-development-ai-navigation-192bdba6fbb4>
- 4 BASLER, Jaromír, 2016. COMPUTER GAMES, THEIR DIVISION, CONTEMPORARY DEVELOPMENT TENDENCIES AND BASIC INVESTIGATIONS FROM THE FIELD OF COMPUTER GAMES. *Trends in Education* [online]. **9**(1), 20-27 [cit. 2024-03-05]. ISSN 18058949. Dostupné z: [doi:10.5507/tvv.2016.003](https://doi.org/10.5507/tvv.2016.003)
- 5 BINARY MOON, 2024. Bone Animation. In: *Brush Ninja* [online]. [cit. 2024-02-06]. Dostupné z: <https://brush.ninja/glossary/animation/bone-animation/>
- 6 COOK, Matt, 2023. How long was Cyberpunk 2077 in development?. In: *Destructoid* [online]. [cit. 2024-01-24]. Dostupné z: <https://www.destructoid.com/how-long-was-cyberpunk-2077-in-development/>
- 7 DE VRIES, Joey, 2024. Deferred Shading. In: *LearnOpenGL* [online]. [cit. 2024-01-24]. Dostupné z: <https://learnopengl.com/Advanced-Lighting/Deferred-Shading>
- 8 DVOŘÁKOVÁ, Lucie, 2023. STRUKTURA HERNÍHO STUDIA. In: *SKVOT* [online]. Dostupné také z: <https://skvt.cz/blog/300-struktura-herniho-studia-lide-a-pozice>

- 9 EPIC GAMES, 2024. Licensing Options. In: *Unreal Engine* [online]. [cit. 2024-01-24]. Dostupné z: <https://www.unrealengine.com/en-US/license>
- 10 EPIC GAMES, 2024. *Unreal Engine 5.0 Documentation* [online]. [cit. 2024-02-02]. Dostupné z: <https://docs.unrealengine.com/5.0/en-US/>
- 11 FILMMAKER, 2023. Unreal Engine Pricing: A Complete Guide to the Cost of UE 5. In: *Filmmaker Tools* [online]. <https://www.filmmaker.tools/> [cit. 2024-01-24]. Dostupné z: <https://www.filmmaker.tools/how-much-does-unreal-engine-cost>
- 12 GREGORY, Jason, 2019. *Game engine architecture*. Third edition. Boca Raton. ISBN 978-1-1380-3545-4.
- 13 GRIFFITHS, Daniel, 2023. Unity reveals Unity 6 and their AI-powered future of games development. In: *PocketGamer.biz* [online]. [cit. 2024-01-24]. Dostupné z: <https://www.pocketgamer.biz/news/82873/unity-reveals-unity-6-and-their-ai-powered-future-of-games-development/>
- 14 H. EBERLY, David, 2005. *3D Game Engine Architecture*. 1. San Francisco: Morgan Kaufmann Publishers. ISBN 0-12-229064-X.
- 15 HAAS, J, 2020. *A History of the Unity Game Engine*. Worcester. An Interactive Qualifying Project. Worcester Polytechnic Institute.
- 16 IVASHCHENKO, Dmitrii, 2023. Unity Realtime Multiplayer, Part 8: Exploring Ready-Made Networking Solutions. In: *HACKERNOON* [online]. [cit. 2024-03-05]. Dostupné z: <https://hackernoon.com/unity-realtime-multiplayer-part-8-exploring-ready-made-networking-solutions>
- 17 JEANNIE, Novak, 2012. *Game Development Essentials: An Introduction*. Delmar: Cengage Learning. ISBN 978-1-1113-0765-3.
- 18 JENSEN, K. Thor, 2024. 25 Years Later: The History of Unreal and an Epic Dynasty. In: *PC Magazine* [online]. [cit. 2024-01-24]. Dostupné z:

<https://www.pcmag.com/news/25-years-later-the-history-of-unreal-and-an-epic-dynasty>

- 19 LEE, Eun-Seok a Byeong-Seok SHIN, 2023. Vertex Chunk-Based Object Culling Method for Real-Time Rendering in Metaverse. *Electronics* [online]. **12**(12) [cit. 2024-01-24]. ISSN 2079-9292. Dostupné z: doi:10.3390/electronics12122601
- 20 MICRO-STAR INT'L. Afterburner. In: *MSI* [online]. [cit. 2024-02-24]. Dostupné z: <https://www.msi.com/Landing/afterburner/graphics-cards>
- 21 MIRROR TEAM, 2023. *Mirror Networking* [online]. [cit. 2024-03-05]. Dostupné z: <https://mirror-networking.gitbook.io/docs/>
- 22 NOBREAKPOINTS, 2024. Unity ProBuilder Guide – So erstellst du deine eigenen 3D-Umgebungen. In: *Nobreakpoints* [online]. [cit. 2024-03-05]. Dostupné z: <https://blog.nobreakpoints.com/unity-probuilder-guide/>
- 23 PHOTON, 2024. *PUN* [online]. [cit. 2024-03-05]. Dostupné z: <https://www.photonengine.com/pun>
- 24 PISKUNOV, Egor, 2021. Comprehensive Comparison of Unreal Engine vs Unity. In: *ILogos Limited* [online]. [cit. 2024-01-22]. Dostupné z: <https://ilogos.biz/comprehensive-comparison-of-unreal-engine-vs-unity/>
- 25 POROKH, Alena, 2023. Top games made in Unreal Engine. In: *Kevuru Games* [online]. [cit. 2024-03-05]. Dostupné z: <https://kevrugames.com/blog/top-games-made-in-unreal-engine/>
- 26 PULSE COLLEGE, 2023. What is game development?. In: *Pulse College* [online]. [cit. 2023-10-20]. Dostupné z: <https://www.pulsecollege.com/what-is-game-development-everything-you-need-to-know/>
- 27 REED, Carol, 2023. Is Unity a Right Choice for Mobile Game Development?. In: *Medium* [online]. [cit. 2024-01-24]. Dostupné z:

<https://medium.com/@carol.reed.597/is-unity-a-right-choice-for-mobile-game-development-b1de47ea0972>

- 28 SALAMA, Ramiz a Mohamed ELSAYED, 2021. A live comparison between Unity and Unreal game engines. *Global Journal of Information Technology: Emerging Technologies* [online]. **11**(1), 01-07 [cit. 2024-01-24]. ISSN 2301-2617. Dostupné z: doi:10.18844/gjit.v11i1.5288
- 29 SATO, Mia, 2023. Epic is changing Unreal Engine's pricing for non-game developers. In: *The Verge* [online]. [cit. 2024-01-24]. Dostupné z: <https://www.theverge.com/2023/10/5/23905082/epic-unreal-engine-pricing-change-film-automotive>
- 30 SAWANT, Vishal a Ajay RANA, 2023. Game Engine Market: Game Engine Market Size, Share & Trends Analysis Report. In: *AI® Market Research and Consulting* [online]. [cit. 2024-01-22]. Dostupné z: <https://www.aimarketreport.com/smart-technologies/game-engine-market>
- 31 SEWELL, Brenden, 2015. *Blueprints Visual Scripting for Unreal Engine: Build professional 3D games with Unreal Engine 4's Visual Scripting system*. 1. Birmingham: Packt Publishing. ISBN 1785286013.
- 32 SCHELL, Jesse, 2008. *The Art of Game Design: A Book of Lenses*. 1. Boca Raton: CRC Press. ISBN 978-0123694966.
- 33 TILTEDU, 2023. Unreal Engine Roadmap 2024: Revolutionizing the Future of Game Development. In: *Medium* [online]. [cit. 2024-01-24]. Dostupné z: <https://medium.com/@tiltedu/unreal-engine-roadmap-2024-revolutionizing-the-future-of-game-development-78de78184ccf>
- 34 UNITY TECHNOLOGIES, 2014. Blizzard goes mobile with its online card battle game, Hearthstone: Heroes of Warcraft. In: <https://unity.com/case-study/hearthstone> [online]. [cit. 2024-03-05]. Dostupné z: <https://unity.com/case-study/hearthstone>

- 35 UNITY TECHNOLOGIES, 2024. *Unity User Manual* [online]. San Francisco: Unity Technologies [cit. 2022-01-24]. Dostupné z: <https://docs.unity3d.com/>
- 36 UNITY TECHNOLOGIES, 2024. Physics solutions for game development. In: UNITY TECHNOLOGIES. *Unity* [online]. [cit. 2024-02-06]. Dostupné z: <https://unity.com/solutions/programming-physics>
- 37 UNITY TECHNOLOGIES, 2024. *Unity User Manual 2022.3 (LTS)* [online]. [cit. 2024-03-05]. Dostupné z: <https://docs.unity3d.com/Manual/index.html>
- 38 UNITY TECHNOLOGIES, 2024. Unity Plans and Pricing. In: *Unity* [online]. [cit. 2024-01-24]. Dostupné z: <https://unity.com/pricing#plans-enterprise>

## 8 Seznam obrázků, tabulek, grafů a zkratk

### 8.1 Seznam obrázků

Obrázek 1 Rozložení trhu herních enginů (Sawant, 2023).....	17
Obrázek 2 Základní komponenty herního enginu (H. Eberly, 2005) .....	18
Obrázek 3 Vizualizace Occlusion culling (Lee, 2023).....	19
Obrázek 4 Znázornění Deferred rendering přístupu (De Vries, 2024).....	20
Obrázek 5 Unity editor stavů animací a přechodů mezi nimi (Unity Technologies, 2024) .....	22
Obrázek 6 Blueprint editor v Unreal Engine (Sewell, 2015).....	24
Obrázek 7 Screenshot Unity verze 0.2b (Haas, 2020).....	26
Obrázek 8 Editor Probuilder otevřený v Unity (Nobreakpoints, 2024).....	37
Obrázek 9 Skript pohybující s testovacími objekty .....	46
Obrázek 10 Vizualizace kolidujících objektů.....	47
Obrázek 11 Ukázka scény s animovanými postavami.....	53

### 8.2 Seznam tabulek

Tabulka 1 Licenční poplatky za využívání Unreal Engine (Epic Games, 2024).....	30
Tabulka 2 Výsledky ostatních měření v testu Unity.....	48
Tabulka 4 Zátěž hardware v průběhu testu v Unreal Engine.....	49
Tabulka 5 Využití hardwarových zdrojů ve statickém testu v Unity .....	51
Tabulka 6 Naměřené hodnoty snímků za sekundu ve statickém testu v Unity .....	51
Tabulka 7 Využití hardwarových prostředků ve statickém testu v Unreal Engine .....	52
Tabulka 8 Využití hardware při testu statického renderování v Unreal Engine.....	52
Tabulka 5 Využití hardwarových zdrojů ve dynamickém testu v Unity .....	53
Tabulka 6 Naměřené hodnoty snímků za sekundu ve dynamickém testu v Unity.....	54
Tabulka 5 Využití hardwarových zdrojů ve dynamickém testu v Unreal Engine .....	54
Tabulka 6 Naměřené hodnoty snímků za sekundu ve dynamickém testu v Unreal Engine .....	54

### 8.3 Seznam grafů

Graf 1 Počet snímků za sekundu v provedených testech v Unity.....	47
---	----

Graf 2 Počet snímků za sekundu v provedených testech v UE .....	49
Graf 3 Využití grafické karty v jednotlivých testech.....	56
Graf 4 Využití procesoru v jednotlivých testech .....	57
Graf 5 Využití systémové paměti v průběhu testů.....	58
Graf 6 Průměrný počet snímků za sekundu v průběhu testů .....	59