



Zadání bakalářské práce

Název:	Identifikace gest a provádění kouzel ve virtuální realitě
Student:	Jakub Vosička
Vedoucí:	doc. Ing. Mgr. Petr Klán, CSc.
Studijní program:	Informatika
Obor / specializace:	Webové a softwarové inženýrství, zaměření Softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	do konce letního semestru 2024/2025

Pokyny pro vypracování

1. Proveďte rešeršní výzkum moderních virtuálních herních aplikací a dalších souvisejících zdrojů na téma provádění kouzel. Inspirujte se zejména Waltz of The Wizards, The Wizards, The Wizards Dark Times a War of Wizards.
2. Uveďte krátce program pro tvorbu 3D modelů Blender a herní engine Unity.
3. Navrhněte a naprogramujte identifikaci gest ve virtuálním prostředí Unity. Použijte přitom programovací jazyk C#.
4. Realizujte alespoň 3 kouzla v prostředí Unity na základě identifikace gest. Vycházejte přitom z analýzy v bodu 1.
5. Realizujte kompletní herní prostředí v Unity s identifikací gest a prováděním kouzel včetně návrhu vhodných modelů postav/monster, terénu a podpůrných 3D modelů.
6. Experimentálně proveďte kvantitativní analýzu spolehlivosti identifikace gest.
7. Testujte herní prostředí provedené v Unity a kvalitativně i kvantitativně herní prostředí srovnajte s výsledky analýzy provedené v bodu 1.
8. Vytvořte dostatečnou dokumentaci.

Bakalářská práce

IDENTIFIKACE GEST A PROVÁDĚNÍ KOUZEL VE VIRTUÁLNÍ REALITĚ

Jakub Vosička

Fakulta informačních technologií
Katedra softwarového inženýrství
Vedoucí: doc. Ing. Mgr. Petr Klán, CSc.
23. října 2024

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2024 Jakub Vosička. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení, je nezbytný souhlas autora.

Odkaz na tuto práci: Vosička Jakub. *Identifikace gest a provádění kouzel ve virtuální realitě*. Bakalářská práce. České vysoké učení technické v Praze, Fakulta informačních technologií, 2024.

Obsah

Poděkování	vii
Prohlášení	viii
Abstrakt	ix
Seznam zkratk	x
Seznam cizích slov a názvů	xi
Cíl práce	1
Úvod	2
1 Analýza	3
1.1 Základní informace o VR	3
1.2 OpenXR	4
1.3 Ovladače	4
1.4 Hra The Wizards – Enhanced Edition	6
1.4.1 Stručný popis	6
1.4.2 Mechaniky kouzel	6
1.4.3 Hodnocení mechanik	10
1.5 Hra The Wizards – Dark Times: Brotherhood	12
1.5.1 Stručný popis	12
1.5.2 Mechaniky kouzel	12
1.5.3 Hodnocení mechanik	17
1.6 Hra Waltz of The Wizard	19
1.6.1 Stručný popis	19
1.6.2 Mechaniky kouzel	19
1.6.3 Hodnocení mechanik	24
1.7 Hra War of Wizards	25
1.7.1 Stručný popis	25
1.7.2 Mechaniky kouzel	25
1.7.3 Hodnocení mechanik	27
1.8 Shrnutí	28
1.9 Výsledek analýzy	29
1.9.1 Základní systém tvorby kouzel	29

1.9.2	Rozšíření tvorba kouzel	29
1.9.3	Interakce kouzel s okolím	30
2	Použité nástroje	31
2.1	Unity	31
2.2	Blender	33
3	Návrh řešení	36
3.1	Herní mechaniky	36
3.1.1	Systém kouzlení	36
3.1.2	Interakce kouzel s okolím	37
3.1.3	Způsob pohybu	37
3.2	Funkční požadavky	37
3.3	Nefunkční požadavky	38
4	Implementace	39
4.1	F1: Vytvoření funkčního herního prostředí	39
4.1.1	Vytvoření a nastavení scény	39
4.1.2	Nastavení ovladačů	41
4.1.3	Pohybový systém hráče	43
4.2	F2: Menu pro nastavením pohybu	45
4.2.1	Vytvoření menu	45
4.2.2	Pozice menu	46
4.2.3	Interakce s menu	47
4.3	F3, F4, F5: Systém tvoření kouzel	48
4.3.1	Design	48
4.3.2	Vytvoření modelů	48
4.3.3	Struktura implementace	49
4.3.4	Popis <i>ScriptableObject</i>	50
4.3.5	Datová struktura pro kouzla a elementy	51
4.3.6	Zpracování vstupů	53
4.3.7	Prohození elementů	55
4.3.8	Detekce gest	56
4.3.9	Výpočet tvaru gest	57
4.3.10	Vzorec nejmenší vzdálenosti bodu od přímky	59
4.3.11	Vytvoření kouzla	60
4.4	F6: Použití kouzel	61
4.4.1	Chování kouzel	61
4.4.2	Princip řízení kouzel	61
4.5	F7: Interakce kouzel s okolím	62
5	Popis navrženého řešení	63
5.1	Popis prostředí (scény)	63
5.2	Ovládání	64
5.2.1	Pohyb hráče	64

5.2.2	Ovládání menu	65
5.2.3	Gesta pro kouzlení	65
5.3	Kouzla	67
5.4	Interaktivní objekty	68
6	Testování	69
6.1	Metodika testování	70
6.1.1	Struktura dotazníku	70
6.2	Kvantitativní analýza	71
6.2.1	Statistika tvorby kouzel	71
6.2.2	Spolehlivost identifikace gest	72
6.2.3	Náročnost na výkon	73
6.3	Kvalitativní analýza	74
6.3.1	Silné stránky navrženého řešení	74
6.3.2	Oblasti pro zlepšení	74
7	Závěr	76
	Obsah příloh	79

Seznam obrázků

1.1	HTC Vive ovladač [3]	5
1.2	Oculus ovladač [3]	5
1.3	Valve Index ovladač [3]	5
1.4	Swordstorm Spell	26
1.5	Imperial Command	26
1.6	Divine Shield Spell	26
2.1	Základní rozmístění oken v Unity (snímek obrazovky) [9]	32
2.2	Základní uspořádání Blenderu (snímek obrazovky) [11]	35
4.1	Struktura objektů	40
4.2	Nastavení <i>XR Origin (XR Rig)</i>	40
4.3	Nastavení XR Controller (Action-based)	41
4.4	Použitý model rukou	41
4.5	Struktura objektů v levém ovladači	42
4.6	Modely, script a složení objektu ruky	42
4.7	Komponenta <i>Locomotion System</i>	43
4.8	Nastavení komponent v <i>XR Origin</i>	43
4.9	Nastavení <i>Continuous Move Provider (Action-based)</i>	44
4.10	Nastavení obou komponent pro rotaci	45
4.11	Vzhled a hierarchická struktura menu	46
4.12	Nastavení komponent pro interakce v menu	47
4.13	Vzhled magických kruhů	49
4.14	Model <i>Gauntlet</i>	49
4.15	Komponenty (Script) <i>MagicController</i> a <i>CrystalController</i>	50
4.16	Ukázka instance <i>ListOfMagic (SOMagicList)</i>	51
4.17	Ukázka instance <i>MagicFire (SOMagic)</i>	52
4.18	Ukázka instance <i>SpellCastShard (SOSpellCast)</i>	52
4.19	Graf stavů a přechodů pro Grip a Trigger	54
4.20	Detekce směru gesta kolem hráče	58
4.21	Zobrazení použití vzorce [15]	59
5.1	Místnost s objekty	63
5.2	Rozvržení ovládání v nárženém řešení	64
5.3	Schématické rozložení Touchpadů pro pohyb	64
5.4	Menu s indikátorem míření	65

5.5	Rozmístění krystalů a gesto pro jejich prohození	66
5.6	Tvary gest kouzel	66
5.7	Ukázka vytvořených kouzel	67
6.1	Graf statistiky tvorby kouzel	71
6.2	Graf přesnosti detekce	72
6.3	Graf s porovnáním FPS	73

Seznam tabulek

1.1	Tabulka kouzel (The Wizards – Enhanced Edition)	7
1.2	Tabulka přechodů stavů	12
1.3	Tabulka kouzel (The Wizards – Dark Times: Brotherhood)	14
3.1	Typy kouzel	37
4.1	Vlastnosti kouzel	61
5.1	Použití kouzel	68

Seznam výpisů kódu

4.1	HandAnimatorController.cs	42
4.2	MenuController.cs	47
4.3	MagicController.cs, detekce gesta prohození krystalů	56
4.4	MagicController.cs, funkce DistPointToLine()	57
4.5	MagicController.cs, vytvoření kouzla	60

Chtěl bych poděkovat především Doc. Ing. Mgr. Petru Klánovi, CSc., za jeho cenné rady, odbornou podporu, věnovaný čas při konzultacích a vedení mé bakalářské práce.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 citovaného zákona.

V Praze dne 23. října 2024

Abstrakt

Cílem bakalářské práce je analyzovat již existujících VR hry, s cílem navrhnout a implementovat mechaniky pro sesílání alespoň 3 kouzel v herním prostředí VR. Práce analyzuje úspěšné VR tituly, jako jsou Waltz of The Wizards, The Wizards, The Wizards Dark Times a War of Wizards. Výsledný interaktivní svět je realizován v herním enginu Unity, využívající jazyk C# a pomocí modelů vytvořených v Blender.

Klíčová slova virtuální realita (VR), magie, VR hra, sesílání kouzel, gestická interakce, Unity Engine, C#, Blender

Abstract

The goal of this bachelor thesis is to analyze existing VR games in order to design and implement mechanics for casting at least 3 spells in the VR gaming environment. The thesis examines successful VR titles such as Waltz of The Wizards, The Wizards, The Wizards Dark Times, and War of Wizards. The resulting interactive world was realized in the Unity game engine, utilizing the C# language and models created in Blender.

Keywords virtual reality (VR), magic, VR game, magic casting, gestural interaction, Unity Engine, C#, Blender

Seznam zkratk

MOBA	online herní bojová PvP aréna (Multiplayer Online Battle Arena)
FPS	snímky za sekundu (Frames per Second)
PvE	hráč proti stvořením (Player vs Entity)
PvP	hráč proti hráči (Player vs Player)
UI	uživatelské rozhraní (user interface)
VR	virtuální realita
SO	scriptable object

Seznam cizích slov a názvů

asset	použitelný prvek do projektu v Unity
crafting	proces výroby nových předmětů z jiných věcí
debugging	proces nacházení a odstraňování závad v kódu
default	normální/základní stav, ve kterém se něco nachází
element	typ magie, živel (led, oheň, voda, ...)
enum	datový typ tvořený konečnou omezenou množinou pojmenovaných hodnot
int	datový typ reprezentující celé číslo
iterace	opakovací cyklus (2 iterace = 2 opakování)
float	datový typ reprezentující desetinné číslo
HTC Vive	jeden ze systému pro používání VR
mana	magická energie
magie	nadpřirozený fenomén, který nejde jinak vysvětlit
prefab	uložená kopie objektu v Unity
preset	předem uložené nastavení komponenty
script	zdrojový kód jako komponent v Unity
scriptable object	statická instance scriptu
Steam	digitální distributor her a dalších aplikací
SteamVR	aplikace pro spouštění prostředí VR přes Steam

Cíl práce

Cílem práce je vytvoření mechaniky pro tvoření kouzel pomocí gest, včetně spustitelného prototypu, ve kterém je možné si kouzla vyzkoušet. Práce vychází z rešeršního výzkumu (analýzy) vybraných VR her. V rešeršním výzkumu se také prozkoumávají problémy, které s vývojem mechanik pro tvoření kouzel souvisí a porovnávají se způsoby, jakými k nim přistupují zkoumané hry. Z těchto výsledků je poté vytvořen nový inovativnější návrh řešení, který přistupuje k této problematice novým způsobem. Tento návrh řešení bude následně implementován do spustitelného prototypu, který obsahuje dle zadání možnost sesílat minimálně 3 kouzla. Prototyp je následně zdokumentován a otestován. Poté jsou výsledky kvalitativně a kvalitativně porovnány s výsledky vybraných her z rešerše.

Úvod

Virtuální realita stále významnějším způsobem vstupuje do oblasti zábavy a interaktivních zážitků. Jeden z důvodů je, že se stává stále přístupnější pro širší veřejnost díky pokroku v technologii a snižování cen. Další je její revoluční přínos ve způsobu prožívání zážitků v herním prostředí, který otevírá dveře novým možnostem interakcí mezi uživatelem a hrou.

Motivací k této práci je právě tento skrytý potenciál, který leží v propojení virtuální reality s magií a gestickým ovládním, místo nudného mačkání tlačítek, což umožňuje ještě větší prožitek a více možností, jak se hráč může vžít do role, kterou herní světy nabízejí. Díky tomu existuje stále rostoucí zájem o herní tituly umožňující hráčům prozkoumávat světy, které normálně nejsou možné, ať se jedná o žánry sci-fi či fantasy. Zároveň kvůli rychlému růstu v technologiích VR, je zde stále prostor pro vytváření nových herních zážitků, jelikož trh ještě nebyl zcela zahlcen. Z tohoto důvodu je hlavním cílem této práce navrhnout a implementovat inovativní řešení sesílání kouzel pomocí gest ve virtuální realitě.

Shrnutí obsahu

Bakalářská práce obsahuje v úvodní části kapitulu *Analýza* s rešeršním výzkumem vybraných her. Následující kapitola *Použité nástroje* popisuje program pro tvorbu 3D modelů Blender a herní engine Unity. Ve třetí kapitole *Návrh řešení* je popsán návrh inovativního řešení systému tvoření kouzel pomocí identifikace gest, umožňujícího realizaci alespoň 3 kouzel. V kapitole *Implementace* je popsána samotná realizace navrženého řešení v prostředí Unity. Výsledek implementace je zdokumentován v kapitole *Popis navrženého řešení*. Testování implementovaného řešení a vybraných her pro porovnání v kvantitativní a kvalitativní analýze je popsáno v kapitole s názvem *Testování*.



Kapitola 1

Analýza

1.1 Základní informace o VR

Technologie VR je tu více než 50 let, ale teprve nyní je její adopce na vrcholu a firmy i jednotlivci ji začínají naplno využívat. Obecně lze říci, že se jedná o headset, který může být buď samostatný (bez PC) nebo připojený k hardwarově silnému PC nebo herní konzoli jako PlayStation 4. Cílem virtuální reality je prezentovat fiktivní digitální svět v pokročilé simulaci včetně snímání pohybu uživatele. Umožnit tak nahlédnout do vzdálených míst naší planety, vyzkoušet extrémní zážitky nebo si zahrát opravdu intenzivní hry či si oddychnout u prostorových filmů.

V dnešní době virtuální realita jakožto technologie již hodně pokročila. Například v lékařství se VR používá na léčbu nemocí, poruch, dokonce i ztráty paměti. Dalším využitím jsou pokročilejší 3D modely pro výuku anatomie nebo plánování operací, dokonce i nácvik operací pro chirurgy. Sportovní trénink ve VR umožňuje replikovat dokonalé podmínky, stejně jako při reálném výkonu činnosti. VR se při vzdělávání nepoužívá jen na univerzitách a školách, ale i ve firmách. Díky simulováním situací ve virtuální realitě, které by jinak byly velice nákladné nebo dokonce nereálné, jsme schopni absorbovat větší množství informací a lépe pochopit souvislosti skrze tento prožitek.

Zábavní průmysl je samozřejmě zatím nejrozšířenější z pohledu využitelnosti VR. Nejrozšířenějším odvětvím jsou hry, které jsou ve společnosti více spojovány s pojmem VR než jakékoliv jiné využití této technologie. Díky VR je dnes možný meeting ve virtuální místnosti s účastníky, kteří se fyzicky nachází v různých částech světa. [1]

1.2 OpenXR

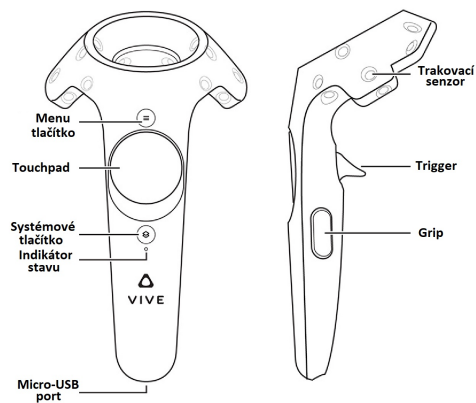
OpenXR je bezlicenční otevřený standard, který poskytuje společnou sadu API pro vývoj XR aplikací, které mohou běžet na široké škále zařízení pro rozšířenou a virtuální realitu. Výrobci nyní mají specifikace, které mohou dodržovat, aby byla zajištěna kompatibilita jejich systému s minulými, současnými a budoucími XR aplikacemi. Vývojáři aplikací se již nemusí starat o cílové platformy, protože OpenXR zajišťuje, že aplikace se bude chovat podobně na všech konformních zařízeních. Spotřebitelé si nyní mohou vybrat svůj preferovaný headset místo toho, aby se rozhodovali na základě kompatibility se svou oblíbenou aplikací. OpenXR podporuje použití široké škály zařízení, a to včetně VR headsetů (Head Mounted Display), ovladačů, základnových stanic, trackovacích senzorů (těla, rukou, objektů, očí ...), haptických zařízení, herních enginů, cloud/5G infrastruktury ... [2]

1.3 Ovladače

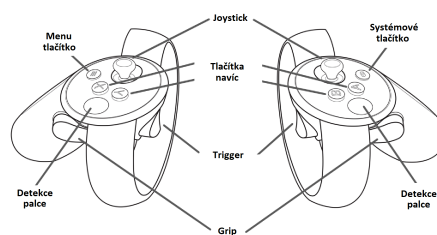
Základní sestava se většinou skládá z headsetu a jednoho ovladače v každé ruce. Existuje několik firem vyrábějících VR systémy, přičemž každá z nich má vlastní typ ovladače. Ovladače jednotlivých firem se mohou lišit, avšak díky standardu OpenXR lze některé společné prvky sjednotit do základní sady ovládacích prvků. Některé ovladače mají dodatečné ovládací prvky, jako je detekování pozice jednotlivých prstů, vibrace či haptická odezva. [3] Společné ovládací, které jsou popsány v následující části, jsou dále v této práci zmiňovány pomocí jejich názvů.

- **Trigger** – Analogová páčka, která detekuje míru stisku, většinou umístěná pro ovládání ukazováčkem. Tvarem i umístěním připomíná spoušť u pistole.
- **Menu** – Tlačítko, které se většinou používá pro spuštění menu ve hře či pro speciální akce.
- **Joystick/Touchpad** – Dotyková plocha nebo klasický joystick s možností mechanického stisku. Většinou je umístěn tak, aby se ovládal palcem a jeho nejčastější použití je pro pohyb ve hře.
- **Grip** – Dvě tlačítka umístěná po obvodu ovladače tak, aby byla zmáčknutelná pevnějším stisknutím ovladače, při jeho klasickém držení.
- **Systém** – Tlačítko, které otevře systémovou nabídku s nastavením VR. [3]

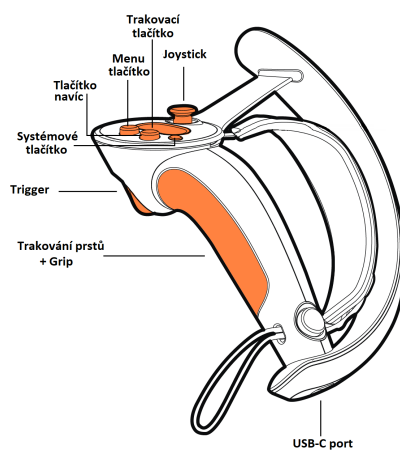
Následující obrázky znázorňují rozložení ovládacích prvků 3 nerozšířenějších značek systémů VR pro PC.



■ Obrázek 1.1 HTC Vive ovladač [3]



■ Obrázek 1.2 Oculus ovladač [3]



■ Obrázek 1.3 Valve Index ovladač [3]

1.4 Hra The Wizards – Enhanced Edition

The Wizards – Enhanced Edition je akční, příběhová fantasy hra pro SteamVR vytvořená v roce 2018 společností Carbon Studio s uživatelským hodnocením 75 %. Na platformě Steam se prodalo 50–100 tisíc kopií. Inspirací k dalšímu textu je autorova přímá zkušenost a informace čerpané z těchto stránek [4, 5].

1.4.1 Stručný popis

Ve světě Meliora hráč postupuje společně s tajemným hlasem kouzelníka, který postupně vysvětluje, co se děje a co se musí udělat, aby zachránili svět před jistou zkárou. Postupně lze prozkoumat několik oblastí, každou s několika úrovněmi, ve kterých čelí hráč různým příšerám a řeší logické hádanky. Každý hráč na své cestě používá šest kouzel, která postupně objevuje a průběžně může vylepšovat.

1.4.2 Mechaniky kouzel

Všechna kouzla se odemknou hned po pár úrovních. Každé z kouzel lze třikrát vylepšit, k prvním dvěma vylepšením dojde postupně pomocí Tokenů. Po odemknutí obou vylepšení se zobrazí unikátní výzva pro dané kouzlo; jejím splněním se vylepší potřeť. Pro získání jednoho Tokenu, musí hráč získat 3 Spell Potential body, které získá za plnění cílů v jednotlivých úrovních. Ve většině se cílí na nalezení 3 skrytých magických úlomků a 3 mety s postupně se zvyšujícím Score, které je spočítáno vždy na konci úrovně.

Všechna kouzla lze rozdělit do několika elementů ohnivě/ledové/bleskové/-fyzické. Kolem některých nepřátel může být vzácně magická aura (štít), která ho ubrání od útoku kouzly stejného typu. Ostatní dodatečné efekty útoků kouzel jsou popsány u jednotlivých kouzel. Kromě útočení se kouzla používají i na řešení různých hádanek a dalších překážek v cestě:

- **Truhla** – Má na sobě obrázek kouzla, které se na ni má použít pro její odemčení a získání předmětu.
- **Poničená zeď** – Blokuje průchod, dá se jakýmkoliv kouzlem zničit.
- **Socha** – Střelí na hráče, lze jí jakýmkoliv kouzlem zničit.
- **Vozík** – je možné odstrčit po kolejích zásahem kouzla Pyroblast.
- **Energický kanón** – Vystřelí po nabití kouzlem Lightning Strike.
- **Pochodně** – Lze ohnivými kouzly zapálit a uhasit pomocí Ice Bow.
- **Rostliny a látky** – Lze ohnivými kouzly zapálit a nechat shořet.
- **Bedny, sudy, ...** – Zničitelné předměty jakýmkoliv kouzlem, občas je v nich něco schované.

V následující části je stručná tabulka 1.1 obsahující základní informace o všech kouzlech. Tyto základní informace obsahují název kouzla, jeho stručně uvedené využití. Každé kouzlo je poté podrobněji popsáno v bodovém seznamu pod touto tabulkou.

Název kouzla	Využití
Arcane Shield	obrana
Fireball	útok na dálku, zapálení
Pyroblast	plošný útok
Frost Bow	útok na dálku, zpomalení
Lightning Strike	útok na krátkou vzdálenost, zpomalené
Arcane Missiles	útok na dálku, naváděný

■ **Tabulka 1.1** Tabulka kouzel (The Wizards – Enhanced Edition)

V následujícím bodovém seznamu jsou všechna tato kouzla popsána tak, jak vypadají, jak se používají a jejich jednotlivá vylepšení. Ve většině případů se kouzlo uvolněním nebo opětovným zmáčknutím Triggeru zruší.

■ Arcane Shield

- **Popis** – Magický štít na ruce, umožňující vyblokovat 3 údery nebo projektily, než se rozbije.
- **Použití** – Vytvoří se v levé ruce stiskem Triggeru a pohybem zleva doprava.
- **1. Vylepšení: Deflection** – Nepřátelské střely se odráží zpět.
- **2. Vylepšení: Arcane Bullwark** – Zvýšená odolnost štítu na 6.
- **Výzva** – Zabití nepřítele do 1 sekundy, po vyblokování jeho útoku.
- **3. Vylepšení: Aegis Throw** – Štít se může zahodit na nepřítele, uvolněním Triggeru.

■ Fireball

- **Popis** – Ohnivá koule, kterou se hází.
- **Použití** – Vytvoří se v pravé ruce dlaní dolů, stiskem Triggeru a následným otočením ruky dlaní vzhůru v malém oblouku. Fireball poté lze zahodit mávnutím ruky a uvolněním Triggeru.
- **1. Vylepšení: Ignite** – Fireball zapaluje nepřátele a způsobuje jim tím průběžné poškození.
- **2. Vylepšení: Empowered Fireball** – Při ponechání Fireballu v ruce delší dobu se zvýší jeho poškození a rychlost kterým letí.
- **Výzva** – Zásahem nepřítele dvakrát za jednu sekundu pomocí plně nabitého Fireballu.
- **3. Vylepšení: Hellfire** – Zásahem již zapáleného nepřítele zapálí ostatní nepřátele v blízkém okolí.

■ Pyroblast

- **Popis** – Výbušná ohnivá koule, kterou se hází. Při dopadu vybuchne a poškodí nepřátele v okolí.
- **Použití** – Vytvoří se s oběma rukama před sebou, souběžným stiskem obou Triggerů a pohnutím dolů. Pyroblast se poté zahodit souběžným mávnutím rukou a uvolněním obou Triggerů.
- **1. Vylepšení: Scorching Earth** – Po výbuchu zůstane na chvíli spálené okolí, které dále poškozuje nepřátele.
- **2. Vylepšení: Living Bomb** – Spálené okolí po výbuchu vybuchne ještě jednou po malé pauze.
- **Výzva** – Zabitím 3 nepřátel jedním výstřelem Pyroblast.
- **3. Vylepšení: Inferno** – Když Pyroblast letí vzduchem létají z něho jiskry, které poškozují nepřátele v okolí.

■ Frost Bow

- **Popis** – Ledový luk střílející ledové šípy, kterými zpomalí a poškodí nepřátele.
- **Použití** – Vytvoří se s oběma rukama před sebou, souběžným stiskem obou Triggerů a překřížením rukou. Šípy se získají prázdnou rukou na zádech a stiskem Triggeru, poté natáhnutím luku a vystřelením při uvolnění Triggeru.
- **1. Vylepšení: Frost Shard** – Nepřátelé budou dostávat větší poškození po zásahu.
- **2. Vylepšení: Icy Mist** – Po zásahu země šípem vznikne mlha, která zpomaluje nepřátele.
- **Výzva** – Zabití 2 nepřátel jedním šípem.
- **3. Vylepšení: Ice Volley** – Zásahem rampouchů, které vzniknou nad mlhou, je shodíte na nepřátele a způsobíte jim poškození.

■ Lightning Strike

- **Popis** – Bleskový paprsek sloučený ze dvou menších, z každé ruky jeden. Krátkodobě paralyzuje a způsobí poškození nepřítelům na střední vzdálenost.
- **Použití** – Vytvoří se s oběma rukama u sebe, souběžným stiskem obou Triggerů, posunutím rychle dopředu a uvolněním Triggerů.
- **1. Vylepšení: Chain Lightning** – Blesky přeskakují na blízké nepřátele.
- **2. Vylepšení: Thunderstruck** – Paralyzuje nepřátele během útoku a efekt trvá i po dobu 1 sekundy po útoku.
- **Výzva** – Zabití dvou typů nepřátel pomocí jednoho Lightning Strike.
- **3. Vylepšení: Double Discharge** – Může se mířit každou rukou zvlášť.

■ Arcane Missiles

- **Popis** – Magické krystaly, které po dotyku rukou automaticky letí na nejbližší nepřátele a způsobí jim poškození.
- **Použití** – Vytvoří se s oběma rukama před sebou, souběžným stiskem obou Triggerů, levou rukou nakreslením C zespodu nahoru a pravou rukou přesně obráceně nakreslením vodorovně převrácené C shora dolů. Poté se jich stačí dotknout pro použití.
- **1. Vylepšení: Arcane Bullets** – Dva krystaly budou mít zesílené poškození, jakmile jedno z nich aktivujete dotykem, tento efekt se rozšíří na další krystal. Po dotyku jednoho normálního krystalu je tento efekt zrušen.
- **2. Vylepšení: Arcane Shower** – Přidá se vnitřní kruh s dalšími 5 krystaly.
- **Výzva** – Seslat všechny krystaly v zesílené formě, bez chyby.
- **3. Vylepšení: Arcane Barrage** – Krystaly se po výstřelu rozdvojí.

1.4.3 Hodnocení mechanik

Jednotlivé výhody a nevýhody herních mechanik jsou popsány v následujícím seznamu spolu s poznatky, které je důležité z těchto bodů vyvodit.

■ Výhody

- Způsob vytváření kouzel pomocí gest je jednoduchý na zapamatování a dává hráči pocit, že vážně ovládá magii.
- Nápadité využití magie k interakci s předměty, jako posouvání vozíku Pyroblastem nebo nabíjení energického kanónu pomocí Lightning Striku.
- Možnost zapálení, zpomalení ...nepřátel, zvyšuje možnosti taktizování a přispívá k zábavnosti bojového systému.
- Vylepšování kouzel postupně zvyšuje komplexnost hry, čímž snižuje náročnost osvojení jednotlivých částí kouzel a posiluje pozitivní pocit z pokroku.
- Tokeny dávají důvod snažit se splnit cíle úrovně a projít každou úroveň s co nejlepším hodnocením, což přispívá k zájmu se perfektně naučit všechna kouzla.

■ Nevýhody

- Malá rozmanitost nepřátel způsobí, že už v polovině hry si hráč zvykne na všechny a přestanete je brát jako výzvu.
- Magické interaktivní předměty, které mají jinou interakci než pouze rozbití, jsou ve hře jen vzácně, což je škoda, jelikož pouhé ničení okolí rychle omrzí.

■ Poznatky

- Vylepšování kouzel, pro které je nutné splňovat výzvy pomocí magie a tím motivovat hráče pro prozkoumání všech vlastností jednotlivých kouzel.
- Ovlivňování nepřátel i status efekty, oproti pouhému poškození, významně zvyšují hratelnost.
- Využití zajímavějších interakcí s předměty, než je pouze jejich rozbíjení, snižuje repetitivitu jednotlivých hádanek.

1.5 Hra The Wizards – Dark Times: Brotherhood

The Wizards – Dark Times: Brotherhood je akční, příběhová fantasy hra pro SteamVR vytvořená v roce 2020 společností Carbon Studio s uživatelským hodnocením 77 %. Na platformě Steam se prodalo 20–50 tisíc kopií. Inspirací k dalšímu textu je autorova přímá zkušenost a informace čerpané z těchto stránek [6, 5]

1.5.1 Stručný popis

Ve světě Meliora se šíří tajemný temný mor, jenž mění obyvatele tohoto světa ve stvoření plné nenávisti a šílenství. Hráč spolu s tajemným hlasem kouzelníka musí bojovat proti zapovězeným silám, zběhlým kouzelníkům a dalším stvořením, která se mu postaví do cesty při snaze zabránit konci světa. Postupně prochází několik oblastí a ovládne 11 kouzel, která pomáhají v boji a proti dalším nástrahám.

1.5.2 Mechaniky kouzel

Kouzla se odemykají postupně během hry. Žádné vylepšení kouzel ve hře není. Výjimkou jsou kouzla, které lze vytvořit gestem z jiných kouzel. Jelikož jsou ve hře popsána zvlášť a fungují jinak zcela samostatně, tak se dále započítávají samostatně.

Každé z kouzel způsobuje poškození určitého elementu (led/oheň/arcané/blesk/radiant). Existuje tady několik druhů nepřátel s elementárním typem. Pokud je na ně zaútočeno stejným elementem, je jejich poškození sníženo a nenastanou proto níže popsané stavy. Různé typy elementů mohou ve správné kombinaci ovlivnit nepřátele a udělovat jim dodatečné poškození. Jednotlivé kombinace je možné popsat pomocí několika stavů podle tabulky 1.2. Normal je defaultní stav, ve kterém se všichni nepřátelé nacházejí. Protivník ve stavu jiném se po krátkém čase vrátí zpět do výchozího stavu Normal.

Původní stav	Typ elementu	Výsledný stav
Normal	ledové	Chill
Chill	ledové	Frozen
Frozen	neohnivé	Shattered
Normal	ohnivé	Warm
Warm	ohnivé	Burn
Chill/Frozen	ohnivé	Wet
Wet	ohnivé	Normal
Normal	bleskové	Shocked

■ **Tabulka 1.2** Tabulka přechodů stavů

Většina stavů jsou stavy přechodné, pouze s grafickým efektem (např. Warm – slabě ohnivá aura na povrchu nepřítele), z nichž pouze pár má nějaký efekt, tyto stavy jsou jednotlivě popsány v seznamu:

- **Burn** – Nepřítel dostává poškození v krátkých intervalech.
- **Frozen** – Nepřítel zamrzne do ledové sochy a nemůže se pohnout ani zaútočit.
- **Shattered** – Ledová socha je roztržena a nepřítel instantně zemře.
- **Shocked** – Nepřítel se třese a je znehybněn.

Kromě boje s nepřáteli, mají kouzla využití na překonání různých překážek, hádanek, či rozbíjení různých objektů:

- **Zarostlé křoví** – Spálí se na popel ohnivým kouzlem.
- **Poničená zeď** – Rozboří se pomocí Arcane Pulse.
- **Zrezivělý kov** – Zničí se hozením Arcane Shieldu.
- **Ohnivé trní** – Musí se zmrazit pomocí ledového kouzla a následně roztržít arcane kouzlem.
- **Umbrella krystal** – Nabije se pomocí několika zásahy radiant kouzel.
- **Runy** – Vyskytují se po trojicích poblíž každé ze dvou truhel, které se odemknou jejich zničením. Runa má na sobě znak konkrétního elementu kouzla, které se musí použít na její zničení.
- **Pochodně** – Zapálí se ohnivým kouzlem a uhasí ledovým.
- **Výbušné elementální vázy** – Zásahem jakýmkoliv kouzlem vybuchnou a způsobí plošné poškození svým elementem.
- **Bedny, sudy, ...** – Doplnkové předměty, které se dají zničit všemi kouzly.

V následující části je stručná tabulka 1.3 obsahující základní informace o všech kouzlech. Tyto základní informace obsahují název kouzla, jeho element (typ poškození) a kouzlo, které musí být předem vytvořeno pro jeho vytvoření (pokud žádné nepotřebuje, je tam pomlčka). Každé kouzlo je poté podrobněji popsáno v bodovém seznamu pod touto tabulkou.

Ovládání kouzel je různé (popsané níže), ovšem pro všechny kouzla platí, že se uvolněním tlačítka (Trigger/Grip), kterým se vytvořilo, zruší. Tato mechanika je pro případ, pokud by se hráč rozhodl pro použití kouzla jiného, než má zrovna vytvořené.

Název kouzla	Element	Vytvořené z kouzla
Fireball	ohnivé	–
Magma Burst	ohnivé	Fireball
Arcane Pulse	arcane	–
Arcane Shield	arcane	–
Arcane Beam	arcane	Arcane Shield
Frost Bow	ledové	–
Ice Sabers	ledové	Frost Bow
Lightning	bleskové	–
Storm Nova	bleskové	Lightning
Radiant Shards	radiant	–
Radiant Missiles	radiant	Radiant Shards

■ **Tabulka 1.3** Tabulka kouzel (The Wizards – Dark Times: Brotherhood)

■ Fireball

- **Popis** – Ohnivá koule, která se hází.
- **Využití** – Hozením způsobuje ohnivé poškození.
- **Gesto vytvoření** – Jednou rukou dlaní dolů stlačením Triggeru a následným otočením ruky dlaní vzhůru v malém oblouku.
- **Gesto použití** – Mávnutím ruky, jako při hodů a uvolněním Triggeru. Je možnost Fireball po nadhození zase chytnout stlačením Triggeru.

■ Magma Burst

- **Popis** – Rotující kotouč žhavého magmatu vytvořený rozžhavením Fireballu, který způsobí rozšiřující se trhlinu s výbuchem po aktivaci nárazem nebo manuálně.
- **Využití** – Trhlina působí mírné ohnivé poškození, výbuch způsobí silné ohnivé poškození plošně a zároveň vyhodí nepřátele do vzduchu.
- **Gesto vytvoření** – S vytvořeným kouzlem Fireball a rukama u sebe, stlačením obou Gripů.
- **Gesto použití** – Oddálením rukou od sebe se začne tvořit trhlina směrem, tam kam ukazují ruce. Uvolněním Gripu se výbuch aktivuje aniž by musel narazit.

■ Arcane Pulse

- **Popis** – Tlaková vlna šířící se od ruky.
- **Využití** – Tlaková vlna odhodí nepřátele a působí efektem arcane elementu bez poškození.
- **Gesto vytvoření** – Jednou rukou, stlačením Gripu a pohnutím horizontálně ruky k sobě.
- **Gesto použití** – Pohybem ruky od sebe a uvolněním Gripu. Rychlejší pohyb udělá kouzlo silnější.

■ Arcane Shield

- **Popis** – Magický štít na ruce, kterým se lze bránit.
- **Využití** – Dá se jím vyblokovat 3 údery nebo projektily, než se rozbije. Alternativně se dá zahodit a tím způsobí arcane poškození.
- **Gesto vytvoření** – Jednou rukou stlačit Trigger a pohnout jím ze strany doprostřed.
- **Gesto použití** – Štít pohnout do cesty projektilům nebo úderům pro jejich vyblokování. Štít lze zahodit mávnutím ruky, jako při hodů a uvolněním Triggeru.

■ Arcane Beam

- **Popis** – Magický paprsek, se začne nabíjet jako malá kulička světla vzniklá stlačením Arcane Shieldu, než se aktivuje.
- **Využití** – Všechno co je v cestě paprsku dostává arcane poškození.
- **Gesto vytvoření** – S vytvořeným kouzlem Arcane Shield a rukama u sebe, stlačením obou Gripů.
- **Gesto použití** – Oddálením rukou od sebe se paprsek aktivuje. Pokud se kouzlo nechá po kratší dobu nabít před aktivací, tak se zvýší jeho síla útoku.

■ Frost Bow

- **Popis** – Luk vytvořený z ledu s 5 ledovými šípy.
- **Využití** – Vystřelené šípy způsobují ledové poškození. Po použití všech 5 šípů se luk rozbije.
- **Gesto vytvoření** – S rukama před sebou, stlačením obou Triggerů a posunutím rukou přes sebe do kříže. Po vytvoření není nutné držet Trigger na ruce bez luku (dominantní ruka).
- **Gesto použití** – Stlačením Triggeru dominantní ruky poblíž tětiny se vytvoří šíp. Uvolněním Triggeru po natáhnutí tětiny a zaměřením stejně jako ve skutečnosti, vystřelíte šíp.

■ Ice Sabers

- **Popis** – Šavle (v každé ruce jedna) z ledu, které se složí z materiálu, ze kterého byl Frost Bow.
- **Využití** – Hozením nebo úderem způsobí poškození a instantní zmrazení nepřátel do stavu Frozen. Jakýmkoliv použitím se hned roztříští.
- **Gesto vytvoření** – S vytvořeným kouzlem Frost Bow a rukama u sebe, stlačením obou Gripů s následným odtažením rukou od sebe.
- **Gesto použití** – Máchnutím na nepřítele nebo zahazením mávnutím ruky, jako při hodů a uvolněním Triggeru.

■ Lightning

- **Popis** – Blesky šlehající z rukou.
- **Využití** – Blesky se na krátkou vzdálenost automaticky zaměří a způsobí bleskové poškození nepřátelům.
- **Gesto vytvoření** – Obě ruce jsou u těla s následným pohybem rukou od sebe a stlačením Triggeru.
- **Gesto použití** – Uvolněním Triggeru.

■ Storm Nova

- **Popis** – Kruhovitá blesková aura, co se zvětší kolem hráče.
- **Využití** – Všem nepřátelům, kteří budou v okolí, způsobí bleskové poškození.
- **Gesto vytvoření** – S vytvořeným kouzlem Lightning a rukama u sebe, stlačením obou Gripů.
- **Gesto použití** – Odtážením rukou od sebe.

■ Radiant Shards

- **Popis** – 5 Duhových krystalů se objeví nad klouby.
- **Využití** – Krystaly po vystřelení rozptýlí v horizontální rovině a každý samostatně uděluje radiant poškození, proto je toto kouzlo vhodné především na blízko.
- **Gesto vytvoření** – Jednu rukou na opačném rameni (levá ruka – pravé rameno a obráceně), stlačením Triggeru a posunem ruky před sebe.
- **Gesto použití** – Rychlým pohybem ruky směrem, tam kam chcete a uvolněním Triggeru.

■ Radiant Missiles

- **Popis** – 10 Duhových krystalů levitujících v kruhové formaci před hráčem.
- **Využití** – Každý krystal je po vystřelení automaticky zaměřen na nejbližšího nepřítele (automaticky k němu letí). Po nárazu uděluje radiant poškození.
- **Gesto vytvoření** – S vytvořením kouzlem Radiant Shards a rukama u sebe, stlačením obou Gripů, počkáním na nabití a poté odtažením rukou od sebe. V závislosti na délce nabití se vytvoří 2 až 10 krystalů.
- **Gesto použití** – Dotykem s jednotlivými krystaly, je vystřelíte.

1.5.3 Hodnocení mechanik

Jednotlivé výhody a nevýhody herních mechanik jsou popsány v následujícím seznamu spolu s poznatky, které je důležité z těchto bodů vyvodit.

■ Výhody

- Způsob vytváření kouzel pomocí gest je jednoduchý na zapamatování a dává hráči pocit, že vážně ovládá magii.
- Kombinace kouzel, jako například zmražení a následné roztržení, které se používá na nepřátele i překážky v prostředí (ohnivé trny).
- Využití už vytvořených kouzel pro vytvoření kouzel nových, což je jednodušší na zapamatování, zkracuje dobu potřebnou pro naučení a snižuje zmatení hráče počtem kouzel.
- Možnost použít více kouzel na vyřešení určité situace a rozmanité interakce s okolními předměty.

■ Nevýhody

- Tím, že nejsou implementovány všechny kombinace mezi kouzly, ztrácí systém celistvost. Nevyužívá ani plného potenciálu této mechaniky a díky tomu může být i matoucí.
- Truhly s Runy, jsou výborný nápad, ale chtělo by to přidat další varianty hádanek pro hráče s využitím kouzel, aby se tak rychle neomrzeli.
- Přes jednoduchost a přehlednost gest se někdy v zápalu souboje nepodaří vytvořit chtěné kouzlo. Občas i ve velkém množství případů. Konkrétně se často místo Arcane Shieldu podařilo vytvořit Fireball, což v souboji proti několika nepřátelům může znamenat i smrt.

■ Poznatky

- Kombinace kouzel zvyšuje jejich rozmanitost a využití, ale je potřeba vytvořit všechny možné kombinace.
- Použití vytvořených kouzel pro aplikaci jiného souvisejícího kouzla zjednodušuje učení a zmenšuje obtížnost zapamatování kouzel.
- Důležitost důkladného popisu a přesnosti detekce gest, pro zamezení seslání špatného kouzla, je klíčová.
- Vysoce žádoucí je také připravit více variant hádanek řešených kouzly, protože oživují svět a zábavnost.

1.6 Hra Waltz of The Wizard

Waltz of The Wizard je hra pro SteamVR vytvořená v roce 2020 společností Aldin Dynamics s uživatelským hodnocením 90 %. Na platformě Steam se prodalo až 20 tisíc kopií. Inspirací k dalšímu textu je autorova přímá zkušenost a informace čerpané z těchto stránek [7, 5].

1.6.1 Stručný popis

V této hře nemá hráč žádný konkrétní cíl, ale má volnost zkoumat okolí a experimentovat s různými předměty s cílem řešit výzvy a záhady, které svět nabízí. Hra začíná v nejvyšší místnosti věže, která kdysi patřila mocnému kouzelníkovi a nyní je plná tajemných předmětů, které zde zanechal. Ve věži, v průběhu téměř každé akce, mu radí tajemná lebka jménem Skully, která leží na stole. Později je možné s touto lebkou mluvit a dávat jí povely na ovlivňování prostředí.

Hráč postupuje dále tím, že splní tři úkoly zobrazené na pečeti u vstupu. Poté se pečeť zlomí a hráči se odemkne vstup na dvůr. Stejným postupem se hráči postupně dostanou na nádvoří a nakonec k portálu do Pevnosti. Věž obsahuje 12 kouzel, která si hráč může odemknout přidáním ingrediencí do kotlíku v určité kombinaci. Mimo věž, kde tato síla není blokována, lze ovládat i tajemnou moc přírodní magie (Nature Magic). V Pevnosti se hráči musí vypořádat s mnoha nepřáteli v 40 generovaných úrovních.

1.6.2 Mechaniky kouzel

Hra obsahuje 3 rozdílné mechaniky.

První mechanika zahrnuje 12 lektvarových kouzel, která jsou odemknuta pomocí kombinace ingrediencí ve věži. Jsou zde 3 skupiny ingrediencí (každá obsahuje 2, 2 a 3 ingredience), které lze smíchat v právě 12 kombinacích. Nelze kombinovat více než jednu ingredienci z jedné skupiny, přičemž přidáním všech 3 ingrediencí se kouzlo aktivuje. Po první aktivaci je kouzlo uloženo do barevné koule v polici nalevo od kotle a pro jeho příští aktivaci ji stačí hodit do kotle. Deaktivovat právě aktivní kouzlo lze pomocí přiložení ruky na určené místo na stole s nápisem "Clear Magic".

Následujících 7 kouzel ovlivňuje objekty v okolí pomocí dotyku:

- **Supersize** – Zvětší se.
- **Miniaturize** – Zmenší se.
- **Transmute** – Změní se na motýla.
- **Polymorf** – Změní se na žábu.
- **Firework** – Vyletí do vzduchu a vybuchne jako ohňostroj.

- **Weightless** – Začne poletovat jako bez gravitace.
- **Symphony** – Zahraje se určitá nota, podle objektu.

Zbývajících 5 kouzel popsaných níže, má své specifické použití:

- **Magnetize** – Stlačením Triggeru přitáhne věci k ruce jako magnet a po uvolnění je od sebe odstřelí.
- **Conductor** – Pro vše v okolí přestane platit gravitace, stlačením Triggeru a pohybem ruky se totožně pohnou všechny okolní předměty. Při uvolnění Triggeru budou předměty pokračovat v pohybu, ve kterém se pohybovaly jako bez gravitace.
- **Puppeteer** – Stlačením Triggeru a chycením předmětu se zaznamená jeho pohyb, až do uvolnění, kdy se začne pohybovat stejně zaznamenal od začátku na konec a zpět, stále dokola.
- **Air Drawing** – Pomocí stlačení Gripu se začne kreslit čára do vzduchu až do uvolnění Gripu, kdy se kresba stane objektem. Triggerem lze tyto objekty uchopit i pustit stejně jako ostatní předměty.
- **Fireball** – Stlačením Triggeru se vytvoří ohnivá koule v ruce, poté máchnutím a uvolněním Triggeru lze vyhodit. Po dopadu vybuchne.

Druhá mechanika se nazývá Nature Magic. Uvnitř věže je blokována, ale jinak funguje vždy a všechna její kouzla jsou hned ze začátku odemknutá. Hráč ze začátku neví jak tato kouzla ovládat, pokud hru již nehrál či někde neviděl. Během prozkoumávání Fortress však narazí na plakáty, které mu poskytnou návod, jak s jednotlivé kouzla použít.

■ Spectral Shield

- **Vzhled** – Štít na ruce z malých bílých částic.
- **Použití** – Zabrání útokům a střelám od nepřátel, pokud je zasažen první. Bílé částice zmizí po zásahu štítu ale po chvíli se obnoví.
- **Gesto** – Stlačením Triggeru v určené ruce a posunutím rukou přes sebe do kříže s určenou rukou jako vnější, uvolněním se zruší.

■ Spectral Blade

- **Vzhled** – Čepel meče vyčnívající z ruky z malých bílých částic.
- **Použití** – Ničí věci a útočí na nepřátele při kontaktu s nimi. Bílé částky zmizí po kontaktu, ale po chvíli se obnoví.
- **Gesto** – Stlačením Triggeru a rychlým posunutím ruky dolů, uvolněním se zruší.

■ Nature Magic

- **Vzhled** – Malé fialové částice poletující kolem ruky.
- **Použití** – Používá se k ostatním kouzlům z přírodní magie.
- **Gesto** – Stlačením Triggeru a posunutím ruky k tělu s dlaní směrem k sobě.

Pro všechna následující kouzla je potřeba nejdříve vytvořit Nature Magic v ruce (v popisu Gesta je myšlena tato ruka).

■ Spectral Orb

- **Vzhled** – Koule z bílých částic, která se chová jako bowlingová koule.
- **Použití** – Koule poškodí všechno do čeho narazí, než za chvíli zmizí.
- **Gesto** – Stlačit Trigger s rukou dole a pohnout jí jako při hodu s bowlingovou koulí s uvolněním Triggeru.

■ Sonic Scream

- **Vzhled** – Zvukové vlny vycházející z pusy.
- **Použití** – Odhodí a poškodí nepřátele v blízkosti.
- **Gesto** – Ruku posunout k ústům, dokud hráč ve hře nespolkne fialové částice, poté zakřičet do mikrofonu.

■ Conduct

- **Vzhled** – Shluk fialových částic, které lze ovládat při letu i po dopadu.
- **Použití** – Po zásahu způsobí poškození a oblepí předmět nebo nepřítele, kterými pak lze pohybovat stejně jako u kouzla Conductor. Na těžké předměty bude fungovat gravitace.
- **Gesto** – Stlačit Trigger, pohnout rukou směrem a rychlostí tam, kam mají částice letět při uvolnění Triggeru.

■ Neutron Orb

- **Vzhled** – Házecí koule z rychle pohybujících se červených částic.
- **Použití** – Po vyhození přitahuje všechny okolní předměty, jako miniaturní černá díra a poškozuje všechno v okolí.
- **Gesto** – Dát ruce k sobě, fialové částice začnou zrychlovat svůj pohyb až zčervenají, zahodí se stlačením a uvolněním Triggeru oběma rukama najednou.

■ Neutron Orb Explosion

- **Vzhled** – Exploze zrychlených červených částic.
- **Použití** – Výbuchem poškodí a odhodí všechno v okolí hráče.
- **Gesto** – Dát ruce k sobě, fialové částice začnou zrychlovat svůj pohyb až zčervenají, zahodí se stlačením a uvolněním Triggeru oběma rukama najednou.

■ Tornado

- **Vzhled** – Malé tornádo, které se pohybuje směrem zahození.
- **Použití** – Přitáhne všechno v okolí, přičemž vše uvnitř tornáda poškozuje a postupně odnáší pryč s sebou.
- **Gesto** – Stlačit Trigger, mávat rukou v horizontálních kruzích, až se začne dělat malé tornádo, zahodí se uvolněním Triggeru.

■ Lightning Bolt

- **Vzhled** – Bleskové paprsky, které se dají hodit jako projektil.
- **Použití** – Po vyhození vystřelí projektil z blesků, který poškodí více než normální Spectral Orb.
- **Gesto** – Stlačit Trigger, rychle pohybovat rukami u sebe střídavě nahoru a dolů, než se vytvoří bleskové výboje, zahodí se uvolněním Triggeru.

Třetí a poslední mechanikou je "Aladinův prsten", který dovoluje komunikovat anglickým jazykem a tím dávat povely pomocí rozpoznávání hlasu. Rozpoznává se několik implementovaných frází s několika alternativními tvary, které ovlivňují okolní svět. V případě nerozpoznání fráze se nic nestane.

Jedním z povelů je například aktivace jednoho z 12 lektvarových kouzel slovy: "Give me ...", kde se doplní název kouzla a deaktivace pomocí slov: "Clear magic". Zbytek povelů umožňuje ovládání různých prvků herního světa (měnit gravitaci, velikosti objektů nebo je vytvářet, ...). V následujícím výčtu jsou popsány základní fráze s jejich efektem. Tady je nutné vzít v potaz, že většinu částí frází lze obměnit či pozměnit například "Make me tiny" změnou na "Make me giant" nebo "Make that bouncy" nahrazením za "Make this bouncy".

- "Make me tiny" – Zmenší hráče.
- "Make me healthy" – Uzdraví hráče od všeho obdrženého poškození.
- "Make me incredibly fast" – Zvýší rychlost chůze.
- "Make my magic very fast" – Zrychlí magické projektily a efekty.
- "Make him sound higher" – Zvýší ukazované osobě (hlavně Skully) hlas.

- **"Make time very slow"** – Zpomalí čas, kromě hráče.
- **"Make gravity super strong"** – Zvýší gravitaci.
- **"Make that bouncy"** – Zařídí, aby se předmět na který se hráč dívá nebo ukazuje, odrazil od všeho.
- **"Make this sticky"** – Udělá předmět na který se hráč dívá nebo ukazuje lepkavým.
- **"Make everything normal"** – Všechno se vrátí (jen jejich vlastnosti) do původního stavu.
- **"Create a grenade"** – Vytvoří granát. Další objekty jsou například: Crossbow, Dart, Coin, Table, Lantern, Box, Butterfly ...
- **"Explode this"** – Věc na kterou hráč ukazuje vybuchne.
- **"Pin this"** – Přišpendlí věc, kterou hráč drží, nebo na kterou ukazuje k povrchu za ní.
- **"Glue this together"** – Spojí k sobě dva předměty, které hráč drží v rukách nebo ukazuje.
- **"Reset room"** – Vše vrátí zpět, jako když se hráč poprvé ocitnul v místnosti.

1.6.3 Hodnocení mechanik

Jednotlivé výhody a nevýhody herních mechanik jsou popsány v následujícím seznamu spolu s poznatky, které je důležité z těchto bodů vyvodit.

■ Výhody

- Získání kouzel pomocí kombinace přísad přidává kreativní mechaniku, poskytující další způsob postupu, čímž se prohlubuje herní zážitek.
- Vizualizace Nature Magic pomocí částic umožňuje detailní způsob vizualizace magie a její reakce na hráčovy podněty.
- Gesta pro sesílání Natural Magic jsou správně zpracovaná tak, aby nedocházelo k seslání jiného kouzla, než hráč zamýšlel.
- Možnost ptát se na základní dotazy, dávat povely či tvořit kouzla, umožňuje hráči silnější propojení s tímto světem.

■ Nevýhody

- Přepínání mezi 12 lektvarovými kouzly pomocí slovních příkazů je časově i ergonomicky nepraktické a nelze se na ně spolehnout v soubojích.
- Velmi často se slova nerozpoznají správně, což způsobuje obtíže při situacích, ve kterých je potřeba spolehlivě a rychle reagovat.
- Přesnost ukazovacích zájmen "this"/"that" není moc jasná, jelikož chybí ukazatel, na co hráč ukazuje, a proto se často stává, že hra vybere jiný předmět, než hráč chtěl ovlivnit.

■ Poznátky

- Odemykání kouzel pomocí míchání přísad či jiných surovin je pozoruhodný nápad, poskytující další způsob postupu.
- Vizualizace magie pomocí shluku částic je efektní a zároveň jednoduchá na provedení.
- Možnost interakce hlasem má vysoký potenciál, ale momentálně je vysoce náročná na správné provedení a má vysokou míru nepřesností.

1.7 Hra War of Wizards

War of Wizards je MOBA hra s magií ve VR vytvořená v roce 2018 společností Arcane Miracle Entertainment s uživatelským hodnocením 76 %. Na platformě Steam se prodalo až 20 tisíc kopií. Inspirací k dalšímu textu je autorova přímá zkušenost a informace čerpané z těchto stránek [8, 5].

1.7.1 Stručný popis

Hra se odehrává v režimu 1v1 nebo 2v2 s ostatními hráči online, kdy se každý hráčský tým snaží zničit krystal nepřátelského týmu. Mapa je rozdělena na dvě strany, přičemž každá strana obsahuje vyvýšenou plochu s hráči, krystalem a obrannou věží. Jednotky se postupně vytvářejí z krystalu a směřují k nepřátelské věži a následně ke krystalu, aby je zničily. Hráči mají k dispozici širokou škálu kouzel, která mohou použít k podpoře svých jednotek, oslabení nepřátelských sil nebo přímo k útoku na nepřátelský tým. Každé kouzlo má přitom svůj unikátní účinek, a hráči si musí strategicky vybírat, jaká kouzla použít.

1.7.2 Mechaniky kouzel

Před každým zápasem si hráč vybírá 4 kouzla z vlastněných, která bude moci použít během souboje. Vybraná kouzla se zobrazí na malém podstavci před hráčem, který obsahuje 5 slotů (4 vybrané před zápasem + 1 bonusový). Během hry se může objevit několik padajících run s dalšími kouzly. Po zásahu těchto run magickou střelou se kouzlo na runě přidá do bonusového slotu a hráč jej může používat po zbytek souboje. Pokud hráč zasáhne další runu, staré nahradí novým.

Hráč má v ruce hůlku, kterou může vystřelovat magické střely pomocí zmáčknutí horní části Touchpadu a kreslit magii do vzduchu stiskem a držením Triggeru. Každé kouzlo má obrazec, který musí hráč správně nakreslit, aby bylo vytvořeno. Potvrdit vytvoření lze stiskem a podržením Gripu. Kouzlo se sešle, pokud má hráč dostatek many a obrazec byl nakreslen rozpoznatelně, a jeho síla závisí na kvalitě nakreslení v porovnání s obrazcem kouzla. Hodnotí se zvláště velikost a přesnost, pomocí 4 známek ($S > A > B > C$), které určují sílu dvou různých částí kouzla. Po seslání se kouzlo začne znova nabíjet a několik sekund se nemůže znovu použít. V opačném případě se kouzlo zruší. Mana se dobývá magickou střelou po každém zásahu nepřítele nebo nepřátelských jednotek.

Hra obsahuje 45 kouzel, ze kterých je 15 ihned odemknutých a zbytek lze za peníze vyhrané v soubojích odemknout. Kouzla se dají rozdělit podle použití na útočná (sešlou útok), vyvolávací (vyvolání jednotek) a strategická (léčení, štít, oslepení nepřítele, ...). Jelikož se kouzla od sebe ve způsobu tvoření neliší, v následující části jsou popsána jen 3 kouzla pro ukázkou mechanik.

■ **Swordstorm Spell** (viz obrázek 1.4 napravo)

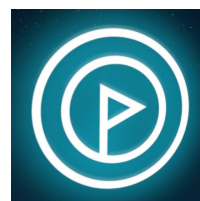
- Cena pro seslání je 400 jednotek many.
- Nabíjecí doba po seslání je 60 sekund.
- Vytvoří **X** magických mečů ve třech salvách, letících na nepřátelský tým.
- Podle přesnosti obrazce každý meč uděluje po zásahu 80/100/120/140 (*C/B/A/S*) poškození.
- Podle velikosti obrazce se vytvoří 48/63/81/99 (*C/B/A/S*) mečů v jedné salvě.



■ **Obrázek 1.4**
Swordstorm
Spell

■ **Imperial Command** (viz obrázek 1.5 napravo)

- Cena pro seslání je 200 jednotek many.
- Nabíjecí doba po seslání je 30 sekund.
- Vyvolá 3 imperiální vojáky, kteří se přidají do bitvy.
- Podle přesnosti obrazce imperiální voják uděluje při útoku 60/65/75/90 (*C/B/A/S*) poškození.
- Podle velikosti obrazce má imperiální voják 700/800/900/1000 (*C/B/A/S*) životů.



■ **Obrázek 1.5**
Imperial Com-
mand

■ **Divine Shield Spell** (viz obrázek 1.6 napravo)

- Cena pro seslání je 200 jednotek many.
- Nabíjecí doba po seslání je 42 sekund.
- Vytvoří magický štít na všech jednotkách v týmu.
- Podle přesnosti obrazce štít ochrání před 500/550/600/700 (*C/B/A/S*) poškození.
- Podle velikosti obrazce štít vydrží po dobu 12/14/16/18 (*C/B/A/S*) sekund.



■ **Obrázek 1.6**
Divine Shield
Spell

1.7.3 Hodnocení mechanik

Jednotlivé výhody a nevýhody herních mechanik jsou popsány v následujícím seznamu spolu s poznatky, které je důležité z těchto bodů vyvodit.

■ Výhody

- Sesílání kouzel kreslením obrázců umožňuje téměř neomezené přidávání dalších kouzel do hry v budoucnosti.
- Omezení počtu použitelných kouzel najednou spolu s rozšířeným celkovým výběrem dělá hru zábavnější a zároveň do ní přidává strategii.
- Zesílení kouzel podle hodnocení ve více metrikách (přesnost + velikost) přidává dynamiku a zesiluje hráče podle jeho dovedností, nejenom zvýšením jejich čísel.

■ Nevýhody

- Způsob sesílání kouzel pomocí kreslení obrázců není vhodný pro akční žánry her, zejména ty, ve kterých je nutné se rychle pohybovat a bojovat proti více protivníkům najednou.

■ Poznátky

- Ovlivnění síly kouzla podle kvality hráčova vstupu ho odměňuje a podporuje snahu o naučení se kouzel.
- Vytvoření více kouzel s omezením počtu zároveň použitelných kouzel zvyšuje možnosti a nezatěžuje hráče více kouzly, než dokáže zvládnout najednou.
- Sesílání kouzel s kreslením komplikovaných obrázců není vhodné pro hry s rychlejším tempem, i když umožňuje téměř neomezené množství kouzel.

1.8 Shrnutí

Každá z her řeší použití gest pro kouzlení unikátním přístupem, který má své silné stránky i slabiny. V této části jsou zkráceně shrnuty poznatky z analyzovaných her.

První hra *The Wizards Enhanced Edition* obsahuje 6 kouzel, které lze postupně vylepšovat. Mezi její přednosti patří jednoduchý bojový systém, společně s různorodým použitím kouzel pro souboj a interakce s předměty. Dále má důmyslný systém vylepšování kouzel s motivací hráče k dosažení maximálního hodnocení v úrovních. Hlavní nedostatky této hry spočívají v nedostatečné rozmanitosti nepřátel a v malém počtu výskytu zajímavých interaktivních předmětů.

V pokračování *The Wizards Dark Times Brotherhood* je celkem 11 kouzel, každé patřící do jednoho z několika elementů. Mezi hlavní výhody patří mechanika vylepšení základních kouzel na kouzla jiná, což umožňuje hráčům snadno zapamatovatelné ovládání. Kombinace elementů při použití kouzel na nepřátele a okolní předměty či možnost šplhání jsou další mechaniky, kterými tato hra vyniká. Avšak nedostatečně využitý potenciál kombinace elementů kouzel a častý problém seslání jiných kouzel, než hráč zamýšlel v rychlém tempu soubojů patří mezi její hlavní nedostatky.

Třetí hra *The Waltz of the Wizards* obsahuje 3 rozdělitelné mechaniky kouzlení: lektvarová kouzla, *Nature magic* (přírodní magie) a hlasové povely. Výhody každé z nich jsou: jednoduché ovládání, dobře navrhnutá gesta s inovativní vizualizací pomocí částic a ovládání svého okolí pouze pomocí hlasových povelů, resp. Nicméně, omezená použitelnost lektvarových kouzel pomocí přepínání hlasovými povely společně s nedostatečně spolehlivým rozeznáním hlasu patří mezi hlavní nevýhody tohoto přístupu.

V poslední hře *War of Wizards* je k dispozici 45 kouzel, která se tvoří kreslením odpovídajících obrazců do prostoru a následným ohodnocením podle jejich velikosti a přesnosti. Výborným nápadem je podle tohoto hodnocení možnost dynamicky vylepšovat různé vlastnosti seslaných kouzel a strategické omezení počtu použitelných typů kouzel pro jeden zápas. Kreslení obrazců pro tvoření kouzel dává téměř nekonečné možnosti přidávat další, avšak tento systém tvoření kouzel není vhodný pro akční hry, je příliš pomalý a nedovoluje volný pohyb při soubojích.

1.9 Výsledek analýzy

Tato kapitola shrnuje klíčové poznatky z celé analýzy, které následně filtruje na ty, které jsou použitelné do návrhu dle autora. Tyto poznatky budou sloužit jako základ pro návrh a implementaci systému kouzlení, který bude kombinovat nejlepší prvky z analyzovaných her a kreativních nápadů autora. Oblasti, které byly v rámci analýzy sledovány jsou základní systém kouzel, rozšíření tvorby kouzel a interakce kouzel s okolím.

V následujících kapitolách jsou u jednotlivých oblastí uvedeny klíčové poznatky pro návrh řešení, vyplývající z výsledků analýzy.

1.9.1 Základní systém tvorby kouzel

Ve všech analyzovaných hrách se dohromady nacházely tři rozlišitelné způsoby tvorby kouzel:

- **Jednoduchá gesta** – Tento způsob se hodí nejvíce k návrhu autora, jelikož umožňuje rychlou reakci v akčních situacích a je snadno zapamatovatelný pro hráče.
- **Kreslení složitých obrazců** – Je naopak pomalé a složité na vytvoření, což je nevhodné při pohybu po scéně.
- **Rozpoznávání hlasu** – Tato technologie ještě není dostatečně spolehlivá a náročná na implementaci.

1.9.2 Rozšíření tvorba kouzel

Základní systém tvorby kouzel lze obohatit pomocí dalších mechanik, které rozšiřují jeho možnosti nebo usnadňují použití. Zde je výčet mechanik z analýzy, které mohou být použity v návrhu:

- **Skládání a kombinování gest** – Přináší větší hloubku a variabilitu herního systému.
- **Dynamické škálování síly kouzel** – Škálování síly kouzel podle přesnosti provedení gesta motivuje hráče ke zdokonalování svých dovedností.
- **Elementární magie** – Přidání typů kouzel rozšiřuje taktické možnosti a přidává novou vrstvu komplexity do soubojového systému.
- **Omezení současně použitelných kouzel** – Toto omezení v kombinaci s vyšším celkovým počtem přidává strategické rozhodování, které kouzla si vybrat v určitých situacích.
- **Postupné odemykání kouzel** – Určuje tempo hry a zabraňuje zahlcení hráče příliš velkým množstvím mechanik najednou.
- **Vylepšování kouzel** – Poskytuje hráčům pocit progresu a slouží jako systém odměn.

- **Tutoriál** – Efektivně seznamuje hráče s mechanikami ve hře, aby je mohl použít.

1.9.3 Interakce kouzel s okolím

Interakce kouzel s herním světem představují další klíčovou mechaniku, která významně ovlivňuje celkový herní zážitek. Analýza odhalila několik důležitých bodů pro implementaci těchto mechanik:

- **Zničitelnost předmětů** – Zničitelné prostředí je v současnosti považováno za nedílnou součást VR her. Tato mechanika zvyšuje míru realismu a interaktivity herního světa.
- **Komplexní interakce** – Zavedení komplexnějších interakcí přispívá k vytvoření bohatšího herního zážitku a rozšiřuje paletu výzev pro hráče. Tyto interakce mohou zahrnovat například řešení hádanek pomocí využití různorodých kouzel nebo manipulace s prostředím (zapálení série pochodní v určitém pořadí).

Použité nástroje

2.1 Unity

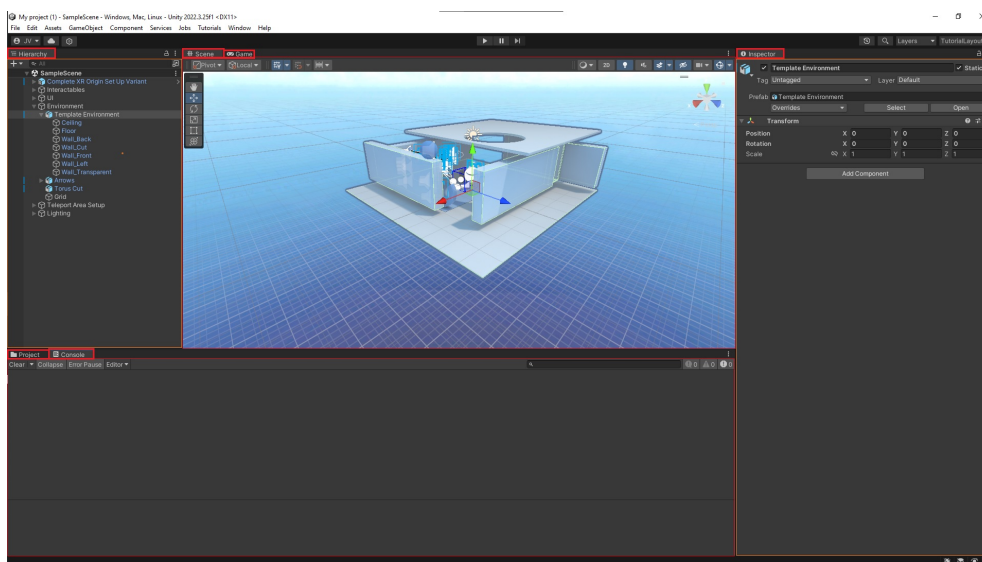
Jde o jeden ze dvou nejznámějších herních enginů společně s Unreal Engine. Unity si získal postavení jednoho z předních herních enginů na trhu, protože zůstává jednoduchý na použití i pro nezkušené uživatele, ale zároveň umožňuje tvorbu s profesionálním vzhledem a funkcionalitou. Jedná se o komplexní nástroj, který sjednocuje mnoho aspektů tvorby digitálních projektů. Poskytuje uživatelům široké spektrum funkcí a nástrojů pro tvorbu, správu a distribuci svých projektů. Unity umožňuje exportovat jednu aplikaci do širokého spektra platforem (PC, mobilních zařízení, herních konzolí, ...) a systémů (Windows, Android, Linux, ...). To umožňuje vývojářům dosáhnout širokého publika bez nutnosti vytváření a údržby oddělených verzí pro každou platformu. K tomu je plně rozšiřitelný pomocí pluginů a rozšíření, což umožňuje například vytvářet programy do VR pomocí rozšíření OpenXR. Kromě těchto rozšíření mohou uživatelé také využívat různé doplňky, knihovny a nástroje třetích stran. [9]

Společně s nejznámější platformou pro tento obsah nazývanou se Unity Asset Store [10], nabízí širokou škálu předem vytvořených modelů, textur, zvukových efektů, skriptů a dalších věcí, které vývojáři mohou využít ke zrychlení svého vývojového procesu. Kromě toho díky rozsáhlé komunitě vývojářů existuje mnoho online fór, komunitních webů, blogů, videí a tutoriálů, kde mohou vývojáři sdílet své zkušenosti, řešit problémy a získávat inspiraci a podporu od ostatních členů komunity. Unity podporuje skriptování pomocí jazyku C# společně s robustním integrovaným vývojovým prostředím (IDE), které zahrnuje funkce jako je syntax highlighting, code completion, debugging a další, což usnadňuje psaní, ladění a správu kódu. Skripty umožňují konfigurovat a ovládat jednotlivé komponenty nebo objekty, protože všechny komponenty a skripty lze inicializovat jako reference, které lze poté přiřadit. [9]

Vyvíjená aplikace se vždy skládá z jedné nebo více scén, ve kterých jsou umístěné jednotlivé objekty. Každý objekt má v základu vždy název, tag,

vrstvu a komponentu Transform určující jeho polohu, rotaci a měřítko. Na objekty se dále přidávají komponenty, které jsou základním stavebním kamenem celého systému, protože určují vlastnosti a chování jednotlivých objektů. Scripty jsou reprezentovány také jako komponenta a společně se širokou škálou dalších komponent, umožňují podporu pokročilého osvětlení, AI, shaderů, animací, fyziky, kolizí, ...Navíc nabízí širokou škálu nástrojů a efektů, které umožňují vývojářům vytvářet vizuálně atraktivní a realistické scény a efekty. V Unity existuje možnost uložit objekt se všemi jeho komponenty a podobjekty jako prefab, který lze použít opakovaně ve všech scénách. To umožňuje snadné vytváření a úpravu opakujících se prvků, což usnadňuje správu a aktualizaci těchto prvků v celém projektu. [9]

Uživatelské rozhraní Unity obsahuje mnoho užitečných nástrojů, jako je například nástroj pro tvorbu terénu, nástroj pro animaci postav, nástroj pro správu zvuků a mnoho dalšího. Tyto nástroje se zobrazují v oknech, která se dají různě přesouvat či skrývat, čímž si každý může přizpůsobit vývojové prostředí dle potřeb (viz obrázek 2.1). Díky drag-and-drop funkci mohou vývojáři jednoduše přidávat, upravovat a propojovat prvky ve svých projektech bez nutnosti hlubšího programování nebo technických znalostí. To umožňuje rychlejší iterace a prototypování, což je klíčové pro úspěch v dynamickém a konkurenčním prostředí herního průmyslu. [9] Základní okna (nástroje) jsou zvýrazněné na následujícím obrázku 2.1 a ve stručnosti popsány v seznamu pod obrázkem.



■ **Obrázek 2.1** Základní rozmístění oken v Unity (snímek obrazovky) [9]

- **Hierarchy** – Zobrazuje všechny objekty v aktuální scéně jako text v hierarchickém seznamu s možností skrytí podobjektů.
- **Inspector** – Zobrazí komponenty zvoleného objektu, které se zde mohou

upravit, přidat či odstranit.

- **Scene** – Vizualizuje rozmístěné objekty v aktivní scéně a umožňuje s nimi pohybovat v dané scéně.
- **Game** – Ukazuje pohled kamery (výstup obrazu/okna vyvíjené aplikace) po spuštění.
- **Project** – Obsahuje složkový systém ve kterém se nalézají všechny objekty, materiály, scripty, modely, prefaby, ...
- **Console** – Vypisuje všechny chybové a informační hlášky při kompilaci a spuštění vyvíjené aplikace. [9]

2.2 Blender

Blender je komplexní 3D grafický software vyvinutý pro tvorbu a animaci trojrozměrných objektů. Jeho univerzálnost a široké spektrum funkcí ho řadí mezi přední nástroje v oblasti digitálního umění a vývoje vizuálního obsahu. Blender vznikl jako interní projekt nizozemské společnosti NeoGeo v roce 1994. Postupem času se stal open-source projektem a v roce 2002 byl uvolněn pod licencí GNU General Public License (GPL). Od té doby prošel mnoha transformacemi a vylepšeními, čímž stal se klíčovým hráčem v oblasti tvorby 3D grafiky a animací. Díky svému otevřenému charakteru a aktivní komunitě vývojářů je Blender nejenom nástrojem profesionálů, ale také oblíbenou volbou pro začínající tvůrce a umělce. Blender umožňuje import a export modelů do různých formátů, což zjednodušuje použití Blenderu s dalšími programy a nabízí širokou škálu nástrojů a funkcí pro modelování, texturování, animaci a renderování, které jsou zde zkráceně popsány. [11]

Mezi hlavní způsoby modelování (vytváření 3D objektů) patří polygonální modelování a sculpting, které společně s několika dalšími umožňují vytvářet různorodé trojrozměrné objekty. Tyto techniky umožňují uživatelům snadno detailně vytvářet složité geometrické tvary a další možnosti úpravy modelu manipulací s jednotlivými částmi modelu, jako jsou hrany, plochy nebo vrcholy. [11]

Blender poskytuje uživatelům nástroje pro práci s tzv. *UV mapou*, texturama a materiálama. *UV mapa* řídí rozložení povrchů 3D objektu na 2D plochu, aby se na něho dala aplikovat textura správným způsobem. Textury se dají vytvářet přímo kreslením na 3D objekt nebo na jeho *UV mapu*. Materiály se tvoří v Shader editoru, který dovoluje textury tvořit pomocí grafického propojování jednotlivých komponent, které různě ovlivňují vlastnosti textur. [11]

Další důležitou částí Blenderu jsou způsoby, jak rozpohybovat modely a vytvářet z těchto pohybů v sekvenci animace. Uživatelé mohou vytvářet kosterní systémy (rig), které umožňují deformaci modelů při pohybu s kostmi, bez pozměnění základní geometrie objektu. Animace se následně tvoří na časové ose, ukládáním rotace, měřítka a pohybu jednotlivých kostí. [11]

Blender také nabízí pokročilé nástroje pro rendering a osvětlení, kterými lze vytvářet fotorealistické snímky, videa a simulace. Umožňuje pracovat s osvětlením, stíny a atmosférickými efekty, pro dosažení až reálných světelných efektů. Obsahuje několik renderovacích enginů, které řídí způsob, jakým se renderují materiály, světelné paprsky a další věci. [11]

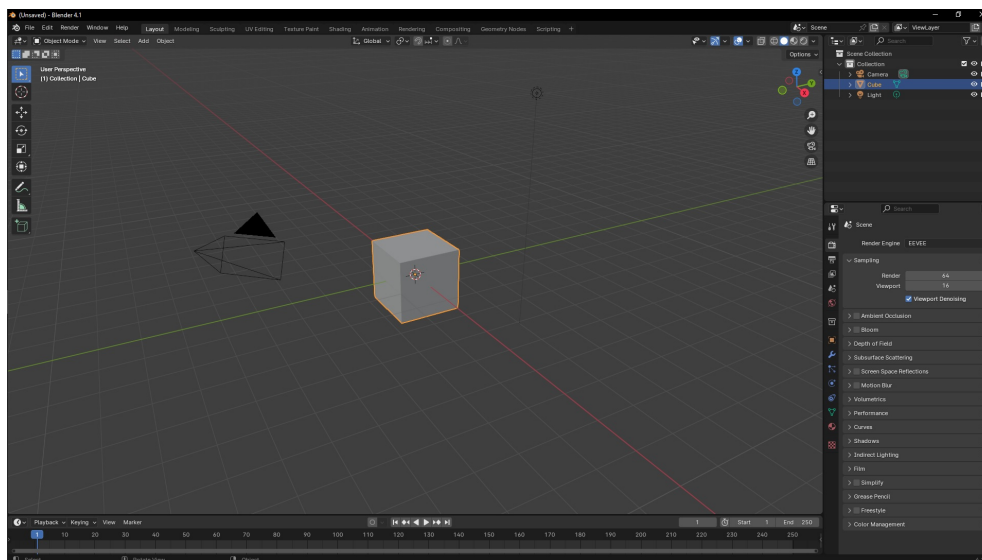
Uživatelské rozhraní Blenderu je tvořeno pomocí několika základních pracovních prostřední (*Workspace*), které se dají přepínat pomocí tabů na horní liště. Každý *Workspace* je rozdělen na několik optimálně rozdělených a nastavených editorů, které dohromady tvoří základní sestavu nástrojů pro danou činnost. Editor má několik režimů v několika kategoriích, které jdou přepínat a každý z nich obsahuje jiné nástroje. Všechny *Workspace* i Editory lze libovolně upravit či přenastavit a upravit si tak tím pracovní prostředí podle potřeb. Pro ukázkou je několik základních *Workspace* ve stručnosti popsáno s jejich účelem, ke kterému jsou přizpůsobené, v seznamu níže. [11]

- **Layout** – Pozicování a úprava celých objektů v prostoru.
- **Modeling** – Vytváření a editování geometrie jednoho objektu.
- **Sculpting** – Editování geometrie jednoho objektu pomocí virtuálních hliněných nástrojů.
- **UV Editing** – Rozmístění textury 3D objektu na 2D plochu.
- **Texture Paint** – Umožní kreslit barvy textury do *UV mapy* nebo přímo na 3D objekt.
- **Shading** – Vytváří různé materiály pomocí jednotlivých modulů a jejich propojením.
- **Animation** – Ukládání momentálních stavů do časové osy s možností editace pozice či dalších údajů jednotlivých objektů. [11]

Téměř každý *Workspace* obsahuje Editor přepnutý do režimu 3D Viewport, který zobrazuje scénu a umožňuje v několika vlastních režimech (Mode) pracovat s objekty (viz obrázek 2.2). Navigace v tomto 3D prostoru je realizována rotací, přibližováním a posouváním pomocí myši a klávesnice. Každý Mode má vlastní využití a umožňuje pracovat na něčem jiném, pro ukázkou jsou níže popsány nejzákladnější režimy, se kterými se téměř vždy pracuje. [11]

- **Object Mode** – Tento režim umožňuje měnit pozici, rozměry, rotaci a další vlastnosti celých objektů, světel, kamer, ...
- **Edit Mode** – V tomto režimu lze vytvářet a upravovat jednotlivé části geometrie objektu (vrcholy, hrany nebo stěny).
- **Sculpt Mode** – Tento režim umožňuje modelování objektů pomocí virtuálních hliněných nástrojů, pomocí tvarování a detailní úpravy povrchů objektů.

- **Weight Paint Mode** – V tomto režimu se přidělují váhy jednotlivým vrcholům v rámci modelu k jednotlivým kostem, což určuje deformaci meshe při animaci.
- **Pose Mode** – Tento režim určený pro animace umožňuje nastavovat pozice a rotace kostí v modelu, čímž mění i geometrii objektu. [11]



■ **Obrázek 2.2** Základní uspořádání Blenderu (snímek obrazovky) [11]

Návrh řešení

V této kapitole je specifikován návrh řešení, sestavený z nápadů autora, výsledků analýzy a zadání. Navrhnuté řešení je zaměřené na systém vytváření kouzel podle gest. Vzhledem k tomuto zaměření není potřeba oproti zadání vytvářet modely postav, monster a složitého terénu. Je potřeba vytvořit pouze minimální herní prostředí a modely související s tvorbou kouzel.

Návrh řešení je rozdělen do několika herních mechanik, ze kterých následně vychází funkční a nefunkční požadavky.

3.1 Herní mechaniky

V této části jsou popsány herní mechaniky, ze kterých se skládá návrh řešení.

3.1.1 Systém kouzlení

Hlavní herní mechanikou je systém vytváření kouzel pomocí gest. Tento systém je unikátní tím, že rozděluje proces vytváření kouzel na dvě části. První částí je výběr elementu a druhou částí je výběr kouzla, které se následně vytvoří ve vybraném elementu. Tento přístup umožní zvýšit celkový počet kouzel tím, že se násobí počtem elementů, aniž by se zvýšil počet gest.

Pro ukázkou této herní mechaniky jsou implementovány 4 typy kouzel (tabulka 3.1) a 3 elementy (oheň/voda/led). Tato kouzla jsou vybrána tak, aby reprezentovala všechny 3 základní kategorie využití v souboji (útok na blízko / útok na dálku / obranu) a 3 typy útoků (sečné/bodné/tupé).

Hráč má možnost na každé ruce individuálně přepínat mezi 3 elementy (1 vybraný + 2 neaktivní). Toto omezení na 3 elementy v každé ruce poskytuje dostatečně velkou paletu možností a zároveň nepřehlčuje hráče moc velkým počtem možností najednou (viz kapitola 1.9).

Kouzlo	Využití	Typ útoku
meč	útok na blízko	sečný
krystal	útok na dálku	bodný
koule	útok na dálku	tupý
štít	obrana	–

■ **Tabulka 3.1** Typy kouzel

3.1.2 Interakce kouzel s okolím

Další herní mechanikou je systém pro interakce objektů s kouzly, který musí být lehce rozšiřitelný.

Pro ukázkou této herní mechaniky je implementováno několik objektů s různým typem interakce na určitá kouzla. Mezi tyto objekty patří pochodeň, která se zapálí nebo uhasí a zničitelná krabice.

3.1.3 Způsob pohybu

Poslední herní mechanika řeší pohyb hráče po prostoru VR. Jelikož je fyzická místnost málokdy dostatečně velká, bývá zvykem umožnit pohyb použitím tlačítek na ovladačích.

Pohyb hráče je nejčastěji realizován plynulým pohybem nebo teleportací (instantním přemístěním). Rotace hráče bývá realizována pomocí plynulého nebo instantního otočení. Všechny tyto způsoby jsou implementovány, kromě teleportace, protože nezapadá do autorova návrhu.

Pro přepínání mezi těmito způsoby pohybu a jejich rychlostmi je vytvořeno menu s možností nastavení pohybu.

3.2 Funkční požadavky

Funkční požadavky specifikují jednotlivé schopnosti a činnosti herních mechanik, popsanych v předchozí části, které musí hra umožňovat. Jejich účelem je rozdělit funkční celek na menší části, které lze poté jednodušeji implementovat.

- **F1: Vytvoření funkčního herního prostředí** – Řešení musí obsahovat základní prostředí ve VR, po kterém se hráč může pohybovat pomocí ovladačů. To zahrnuje základní ovládání chůze a rotace hráče.
- **F2: Menu s nastavením pohybu** – Musí být implementováno uživatelsky přívětivé menu, ve kterém si hráč může nastavit různé možnosti pohybu a rotace.
- **F3: Zvolení elementu** – Hráč musí mít možnost zvolit si element na každé ruce nezávisle. Každá ruka musí umožňovat přepínání mezi třemi elementy (jeden aktivní, dva neaktivní).

- **F4: Rozpoznání gesta** – Řešení musí být schopen rozpoznat gesta provedená hráčem a určit, které kouzlo se má vytvořit.
- **F5: Vytvoření kouzla** – Detekované kouzlo se musí vytvořit pomocí vybraného elementu.
- **F6: Použití kouzel** – Kouzla vytvořená hráčem musí být použitelná v herním světě. Jejich použití se liší na základě typu kouzla a elementu.
- **F7: Systém interakcí kouzel s objekty** – Musí být implementován systém, který umožní interakci kouzel s objekty v herním světě.

3.3 Nefunkční požadavky

Nefunkční požadavky je odborný termín pro technické parametry, které se zaměřují na kvalitativní aspekty, jako jsou výkon, spolehlivost a rozšiřitelnost. Jejich účelem je jasně a srozumitelně specifikovat s čím se má počítat při vývoji.

- **N1: Stabilní aplikace s min 90 FPS** – Aplikace musí běžet stabilně na testovací sestavě s minimální obnovovací frekvencí 90 snímků za sekundu, aby byl zajištěn plynulý a příjemný zážitek ve virtuální realitě.
- **N2: Úspěšnost rozpoznání gesta nad 70 %** – Řešení musí být schopno rozpoznat správně gesta provedené hráčem v alespoň 70 % případů, aby byla zajištěna dostatečná spolehlivost a plynulost herního zážitku.
- **N3: Snadná rozšiřitelnost počtu a vlastností elementů** – Řešení musí být navrženo tak, aby bylo možné snadno přidávat nové elementy a rozšiřovat jejich vlastnosti bez nutnosti zásadních úprav kódu.
- **N4: Snadná rozšiřitelnost počtu a vlastností kouzel** – Podobně jako u elementů, musí být systém schopen snadno přidávat a rozšiřovat kouzla, což zahrnuje přidávání nových typů kouzel a rozšíření jejich chování.
- **N5: Snadné rozšíření interakcí kouzel s prostředím** – Řešení musí být navrženo tak, aby bylo možné jednoduše přidávat nové typy interakcí mezi kouzly a objekty v herním prostředí. To zahrnuje schopnost definovat nové objekty reagující na specifická kouzla bez nutnosti zásadního přepracování stávajícího kódu.

Implementace

Kapitola číslo 4 se věnuje popisu implementace řešení, které vychází z funkčních a nefunkčních požadavků definovaných v kapitole 3 (Návrh řešení). Názvy podkapitol obsahují zkratku funkčního požadavku s číslem, která je identifikuje.

Pro implementaci řešení je využit herní engine Unity (verze 2022.3.25f1) s aplikací šablony *VR (Core)*. Ta poskytuje předinstalované rozšíření *OpenXR*, které zajišťuje kompatibilitu s různými systémy VR.

Projekt vytvořený z této šablony v základu obsahuje scénu a assety, které ukazují možné použití *OpenXR*. Z těchto assetů jsou použity pouze některé části pro UI.

4.1 F1: Vytvoření funkčního herního prostředí

Tato kapitola se zabývá vytvořením kompletního herního prostředí a reprezentace hráče ve VR. V následujících kapitolách je popsán postup vytvoření a nastavení scény, nastavení ovladačů a vytvoření pohybového systému hráče.

4.1.1 Vytvoření a nastavení scény

V této kapitole je nejdříve vytvořena nová scéna, která v Unity reprezentuje herní prostředí a obsahuje všechny objekty. Ta se následně upraví, aby podporovala spuštění ve VR a správně používala všechny základní části VR (Headset, Ovladače).

Nově vytvořená scéna obsahuje dva objekty *Main Camera* (definuje pohled do aplikace) a *Directional Light* (vytváří globální osvětlení).

Pro zprovoznění rozhraní VR je nutné nahradit *Main Camera* objektem typu *XR Origin (VR)*, který vytvoří objekty *XR Origin* a *XR Interaction Manager* (viz obrázek 4.1). *XR Origin* reprezentuje hráče s jeho VR rozhraním (headset, levý a pravý ovladač). *XR Interaction Manager* v sobě obsahuje stej-

nojmenný komponent, který umožňuje fungování systému interakcí (obsahuje souhrn funkcionalit z *Open XR*).

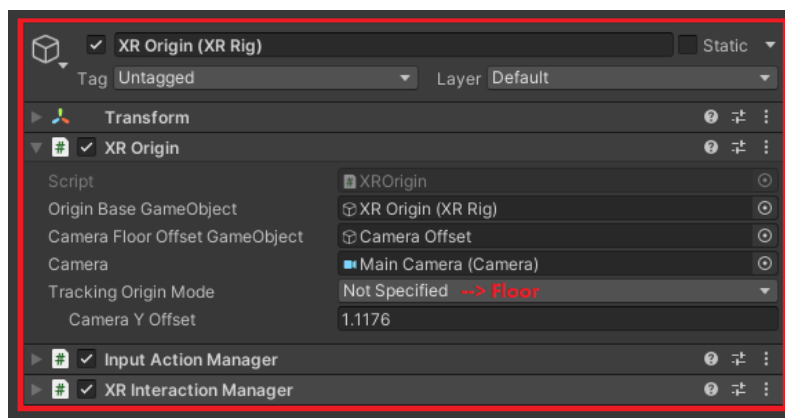


■ **Obrázek 4.1** Struktura objektů

Dalším krokem je vytvoření prostředí, ve kterém se hráč může pohybovat. Toto prostředí je jednoduchá čtvercová místnost vytvořená ze 6 3D objektů *Plane*, který slouží jako podlaha, 4 zdi a strop.

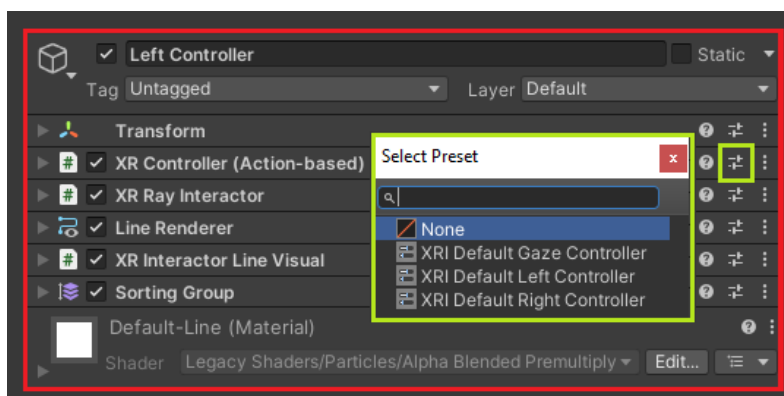
V objektu *XR Origin* na stejnojmenné komponentě je důležité nastavit proměnnou *Tracking Origin Mode* na *Floor*, aby se výška kamery (hráče) počítala podle nastavení konkrétního VR systému (údaj z nastavení VR headsetu).

Pro zpřehlednění je komponenta *XR Interaction Manager* přesunuta na objekt *XR Origin (XR Rig)* (viz obrázek 4.2).



■ **Obrázek 4.2** Nastavení *XR Origin (XR Rig)*

Na obou ovladačích je potřeba nastavit komponentu *XR Controller (Action-based)* na její příslušný preset (viz obrázek 4.3). Tyto presety definují, jak se namapuje sledování (*Tracking*) a ostatní vstupy pro správnou funkci ovladačů.

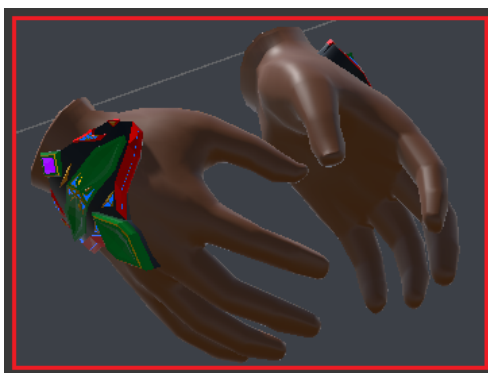


■ Obrázek 4.3 Nastavení XR Controller (Action-based)

4.1.2 Nastavení ovladačů

Dalším krokem je nastavení modelu ovladačů a zprovoznění animaci při zmáčknutí tlačítek.

Použitý model je sestavený z volně dostupné ruky od Oculus [12] a modelu *Gauntlet*, který vytvořil autor v Blenderu (viz obrázek 4.4). Proces vytvoření tohoto modelu je uveden v kapitole 4.3.2.



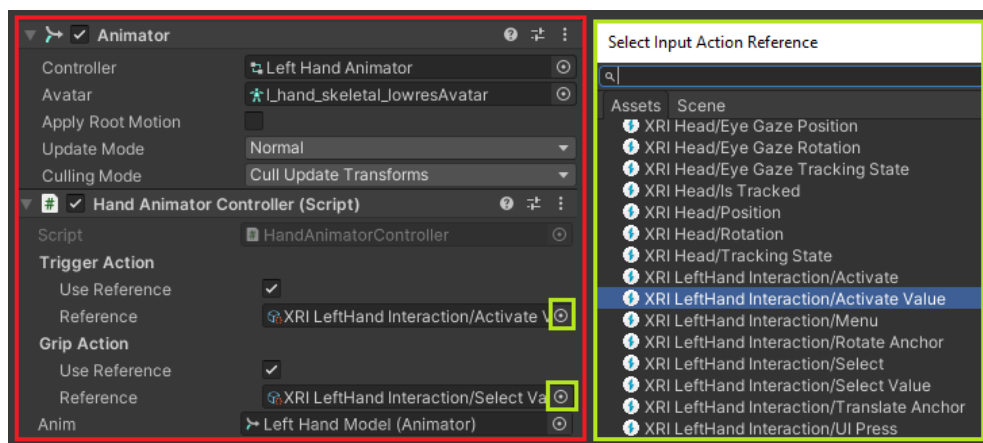
■ Obrázek 4.4 Použitý model rukou

Pro nastavení tohoto modelu bylo potřeba v obou ovladačích udělat následující kroky. Nejdříve smazat výchozí model, který se nachází v proměnné *Model Prefab* na komponentě *XR Controller (Action-based)*. Následně vložit vytvořený model jako podobjekt do ovladačů (viz obrázek 4.5) a nastavit se reference na něj přetažením do proměnné *Model Transform* v komponentě *XR Controller (Action-based)*.



■ **Obrázek 4.5** Struktura objektů v levém ovladači

Pro použití již obsažené animace pohybu prstů, bylo potřeba vytvořit skript s názvem *HandAnimatorController* (viz výpis z kódu 4.1). V něm se využívá datový typ *InputActionProperty*, do kterého lze vložit reference na vstup z OpenXR (viz obrázek 4.6). Z této reference lze následně číst hodnota, v tomto případě typu *float* (0.0 až 1.0, míra stisku), která přímo nastavuje parametry v referenci na animátor.



■ **Obrázek 4.6** Modely, script a složení objektu ruky

```

1 public class HandAnimatorController : MonoBehaviour {
2     [SerializeField] InputActionProperty triggerAction;
3     [SerializeField] InputActionProperty gripAction;
4     [SerializeField] Animator anim;
5     void Update(){
6         anim.SetFloat("Trigger", triggerAction.action.
7             ReadValue<float>());
8         anim.SetFloat("Grip", gripAction.action.ReadValue<
9             float>());
10    }
11 }

```

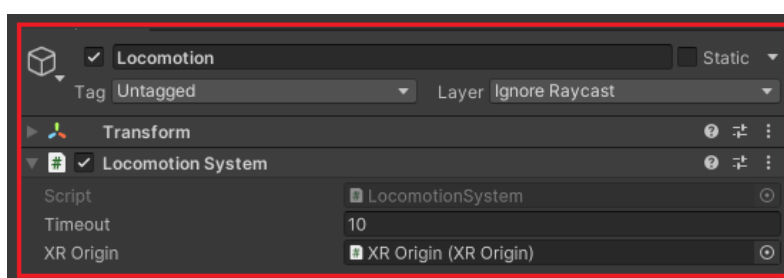
■ **Výpis kódu 4.1** HandAnimatorController.cs

4.1.3 Pohybový systém hráče

Pohybový systém hráče bývá ve většině VR her realizována pomocí Touchpadu. Levým se ovládá chůze a pravým se ovládá otáčení. V této práci je použit stejný způsob ovládání.

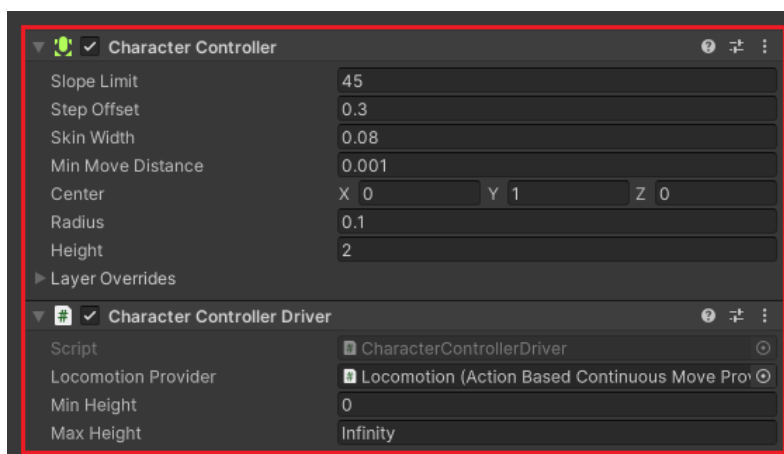
Pro vkládání všech potřebných komponent pro pohybový systém hráče, je potřeba nejdříve vytvořit prázdný objekt pojmenovaný *Locomotion*, jako podobjekt do *XR Origin*.

První komponenta se jmenuje *Locomotion System*. Tato komponenta je hlavní částí, která umožňuje tento pohybový systém (viz obrázek 4.7).



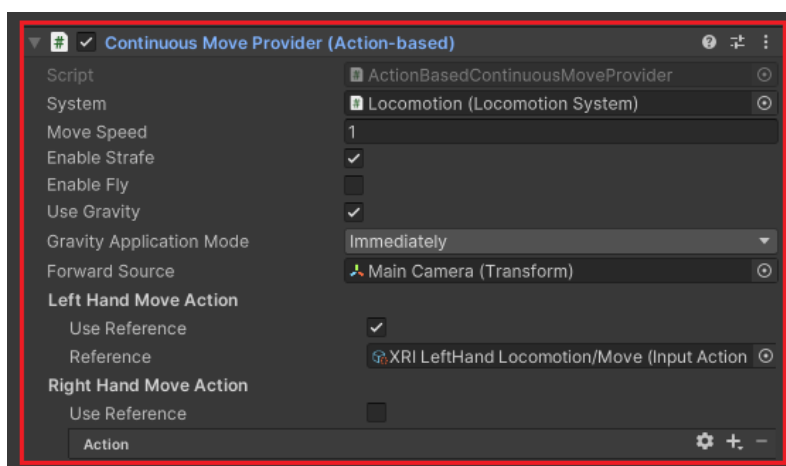
■ Obrázek 4.7 Komponenta *Locomotion System*

Pro správnou funkci této komponenty je ještě přidat dvě další do objektu *XR Origin*. První vytváří kolizní zónu pro tělo hráče (*Character Collider*), ve kterém se upravila hodnota *Radius* na 0.1 – určující šířku hráče a druhá komponenta (*Character Collider Driver*) tuto kolizní zónu dynamicky transformuje (viz obrázek 4.8).



■ Obrázek 4.8 Nastavení komponent v *XR Origin*

Druhou komponentou přidanou do objektu *Locomotion* je *Continuous Move Provider (Action-based)*, která umožňuje plynulou chůzi. V ní bylo potřeba přenastavit několik parametrů a referencí (viz obrázek 4.9), které jsou popsány v seznamu pod obrázkem.

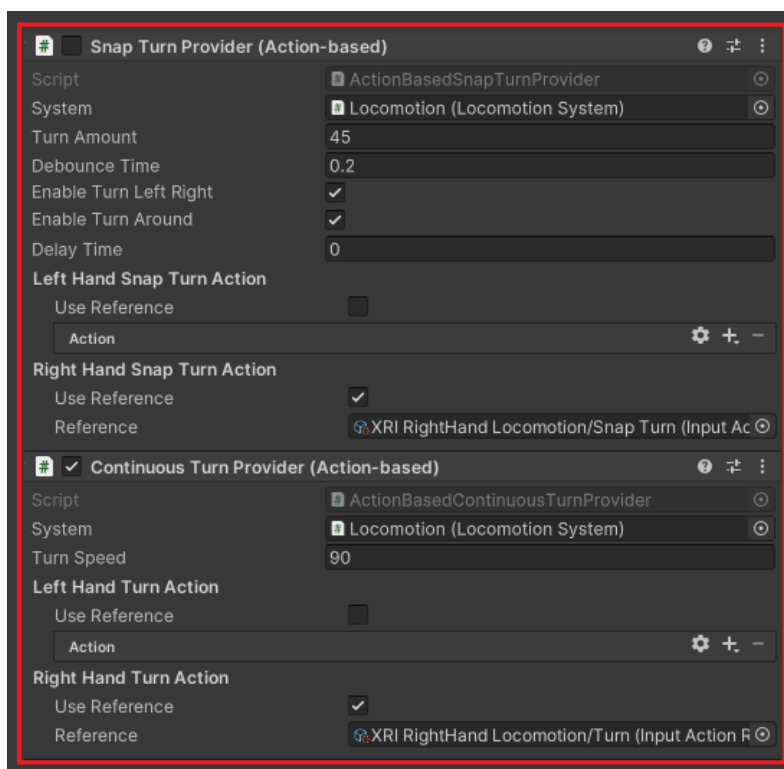


■ **Obrázek 4.9** Nastavení *Continuous Move Provider (Action-based)*

- Reference na *Locomotion System* do proměnné *System*
- Reference na vstup Touchpadu levého ovladače do *Left Hand Move Action*
- Přepnutí *Gravity Application Mode* na *Immediately*
- Reference na *Main Camera* do *Forward Source*, pro určení orientace chůze podle rotace headsetu

Plynulá rotace je realizována komponentou *Continuous Turn Provider (Action-based)* a instantní rotace komponentou *Snap Turn Provider (Action-based)*. Obě tyto komponenty byly přidány do objektu *Locomotion* a v každé byla nastavena reference *Right Hand Move Action* na vstup Touchpadu z pravého ovladače.

Jelikož může být zapnutý pouze jeden způsob rotace najednou, je potřeba jednu z nich deaktivovat (viz obrázek 4.10).



■ Obrázek 4.10 Nastavení obou komponent pro rotaci

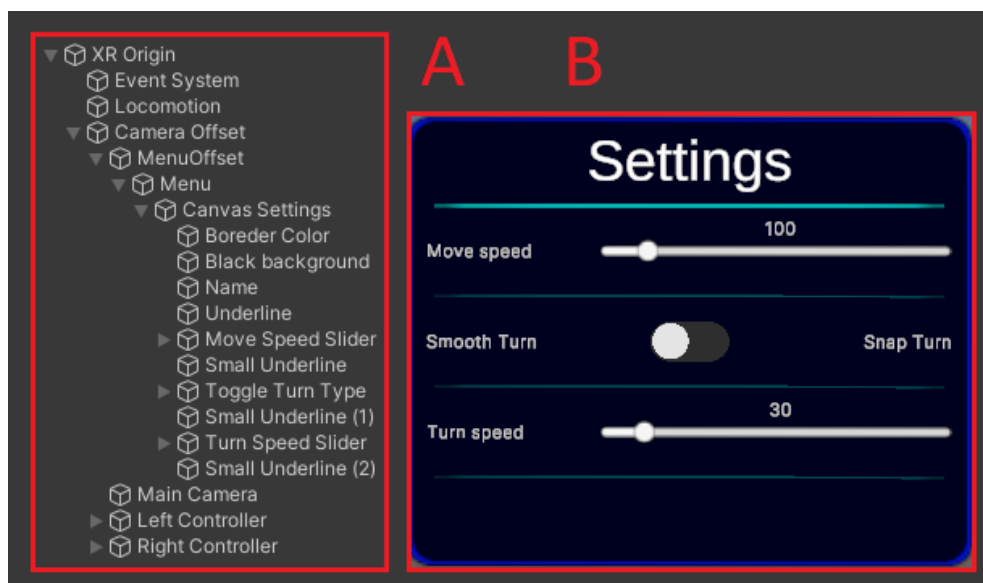
4.2 F2: Menu pro nastavením pohybu

Tato kapitola se zabývá implementací menu pro nastavením pohybu hráče. V následujících podkapitolách se řeší vytvoření tohoto menu, jeho pozice ve scéně a interakce s ním.

4.2.1 Vytvoření menu

Menu se vytváří pomocí speciálního objektu *UI Canvas* z *OpenXR*. Do tohoto objektu se poté vkládají interaktivní prvky (posuvník, přepínač, text, ...). Při vytvoření *UI Canvas* se zároveň vytvoří i objekt *Event System*. Tento nový objekt řídí interakce s objekty v *UI Canvas*.

Pro přehlednou organizaci byla vytvořena hierarchie prázdných objektů *Menu > MenuOffset* a byly vloženy jako podobjekt do *Camera Offset*. Dvnitř *MenuOffset* je umístěn dříve vytvořený objekt *UI Canvas*, který byl přejmenován na *Canvas Settings*. *Event System* se přemístí do *XR Origin* (viz obrázek 4.11 část A).



■ **Obrázek 4.11** Vzhled a hierarchická struktura menu

Objekt *Menu* je přemístěn na pozici *Camera Offset* a *MenuOffset* je napolozicován tak, aby *Canvas Settings* směřoval *UI Canvas* na hráče ve správné vzdálenosti.

Pro vytvoření interaktivních prvků jsou použité lehce upravené assety ze šablony (*VR (Core)*), které jsou vloženy do objektu *Canvas Settings* (viz obrázek 4.11 část B). Všechny tyto nově vytvořené objekty byly nastaveny do vrstvy (*Layer*) *UI*, aby se poté daly snadněji nastavit interakce (viz kapitola 4.2.3).

4.2.2 Pozice menu

O správné umístění menu do zorného pole hráče se stará script *MenuController*, který je vložen do *Menu*. Kromě této funkce ještě detekuje stisknutí tlačítka *Menu* na jakémkoliv ovladači, pro otevření/zavření menu (aktivuje/deaktivuje objekt).

V Unity si podobjekty udržují stálou relativní pozici a rotaci k výchozí rotaci nadřazeného objektu. Díky tomu stačí posunout nebo otočit nadřazený objekt a všechny pod ním se pohnou také. Tohoto principu se využívá pro posouvání a rotaci objektu *Menu*, což posouvá i vytvořené menu.

K úpravě pozice a rotace se využívá funkce *Lerp*, jenž zajišťuje hladký přechod v čase na základě rozdílu mezi současným stavem a cílovou hodnotou (viz výpis z kódu 4.2). V editoru Unity osa *Y* reprezentuje výšku (vertikální směr) a zbývající osy *X*, *Z* reprezentují šířku a hloubku (horizontální směry). Pozice *Menu* se neustále mění (pokud je otevřené), ale pouze v osách *X* a *Z* na pozici *Main Camera*, jelikož výška od podlahy (osa *Y*) se automaticky předává

z nadřazeného objektu *Camera Offset*. Rotace *Menu* se zapíná v případě, že je rozdíl úhlu pohledu hráče s pozicí menu je větší než 30° , a vypíná v případě, že je tento rozdíl menší než 1° .

```

1 //plynulé posunutí v ose X, Z
2 transform.position = Vector3.Lerp(transform.position,
   new Vector3(headset.position.x, transform.position.y,
   headset.position.z), Time.deltaTime * lerpSpeed);
3 //plynulá rotace v ose Y
4 transform.rotation = Quaternion.Lerp(transform.rotation,
   Quaternion.Euler(0f, headset.rotation.eulerAngles.y,
   0f), Time.deltaTime * lerpSpeed);

```

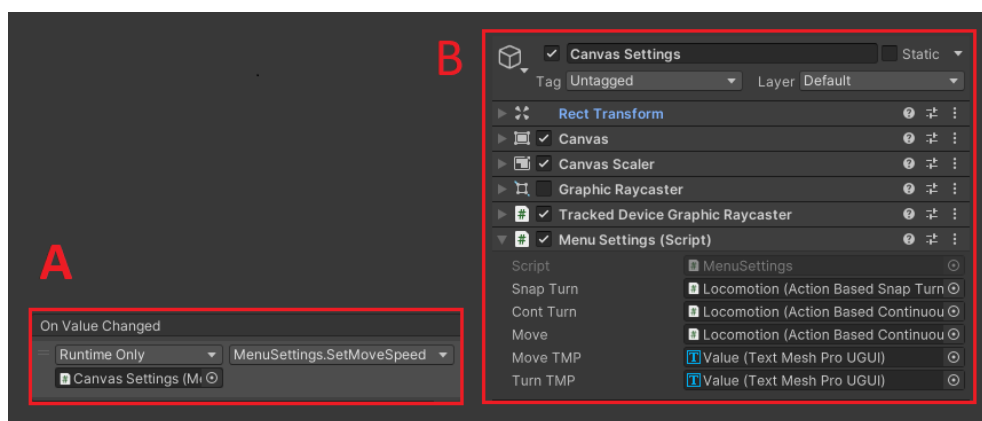
■ **Výpis kódu 4.2** MenuController.cs

4.2.3 Interakce s menu

Pro správné ovládání menu hráčem, je potřeba nastavit na každém ovladači komponentu *XR Ray Interactor* tak, aby reagovala pouze na vrstvu (*Layer*) UI a schovávala mířící ukazatel, pokud na ni nemíří.

Tyto interakce jsou řízeny přímo pomocí vbudovaného systému v jednotlivých prvcích menu (posuvník, přepínač, ...), na jejichž hlavní komponentě se dá nastavit zavolání funkce ze scriptu, pokud dojde ke změně jejich stavu (viz obrázek 4.12 část A).

Script *MenuSettings*, který je přidán na objekt *Canvas Settings* obsahuje všechny tyto volané funkce a reference na komponenty, které tyto funkce mají nastavovat (viz obrázek 4.12 část B).



■ **Obrázek 4.12** Nastavení komponent pro interakce v menu

4.3 F3, F4, F5: Systém tvoření kouzel

Tato kapitola se zabývá implementací systému tvoření kouzel, který se skládá ze tří mechanik: výběru elementu, rozpoznání gesta a vytvoření kouzla. Vzhledem k úzké provázanosti těchto tří mechanik jsou v této kapitole považovány za jednotný funkční celek. Tento systém umožňuje nezávisle na druhé ruce (ovladači) si zvolit element, vytvořit a rozpoznat gesto, pomocí kterého se poté v kombinaci se zvoleným elementem vytvoří kouzlo.

4.3.1 Design

Tato část se zabývá klíčovými designovými rozhodnutími, která jsou zde zkráceně popsána pro přehlednost.

Elementy jsou zastoupeny barevnými krystaly na každé ruce hráče. Každá ruka má tři místa (sockety) pro krystaly, jeden je umístěn uprostřed a dva po stranách. Aktivní (používaný) element se určuje podle krystalu v prostředním socketu. Každá ruka může mít své vlastní rozvržení krystalů nezávisle na ruce druhé.

Výměna krystalů probíhá jednoduchým gestem, které prohodí jeden z krystalů po straně (dle volby hráče) s prostředním. Toto gesto, které je níže popsáno, lze přerušit uvolněním tlačítka Grip nebo nedodržením směru (krok 2).

1. Stisknutí tlačítka Grip.
2. Posun ruky vodorovně podél směru dlaně k hráči.
3. Posun ruky směrem do jednoho ze 2 magických kruhů (vzhled dle obrázku 4.13), které se objeví po splnění předchozího kroku (každý nich reprezentuje jeden postraní krystal).

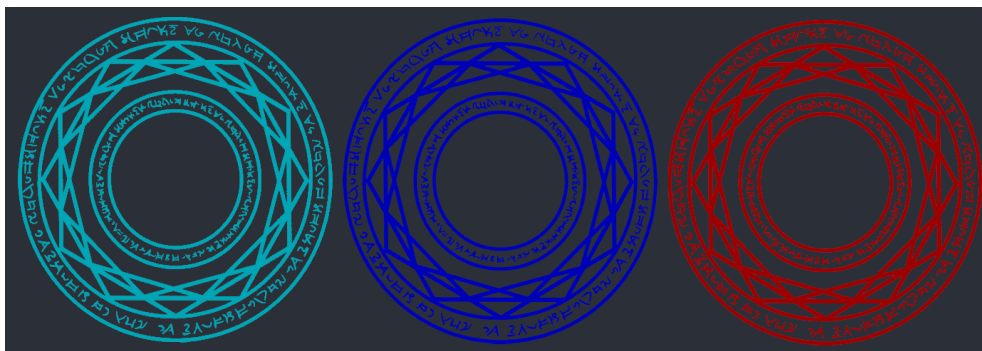
Kouzla se tvoří nakreslením specifického tvaru ve vzduchu, tento tvar je při kreslení viditelný jako čára v barvě vybraného elementu. Kreslení probíhá v 3D prostoru při držení tlačítka Trigger. Po uvolnění tlačítka se vyhodnotí nakreslený tvar. Pokud odpovídá definovanému kouzlu, tak se vytvoří v momentálně vybraném elementu kreslicí ruky.

4.3.2 Vytvoření modelů

V této části je vysvětlen postup při vytvoření modelů kouzel, magických kruhů a *Gauntlet*, který zobrazuje elementy na obou ovladačích (viz kapitola 4.1.2).

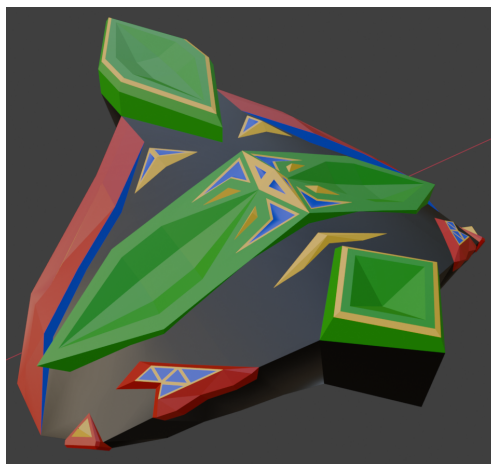
Modely kouzel byly vytvořeny autorem v programu Blender formou jednoduchých a výstižných modelů (meč/koule/krystal/štít). Za zmínku stojí hlavně jejich textury, pro které byly použity volně stáhnutelné *shader masky* [13].

Magické kruhy (viz obrázek 4.13) jsou vytvořeny jako obyčejné objekty *Plane* s poloprůhlednou texturou [14] volně dostupnou na internetu.



■ Obrázek 4.13 Vzhled magických kruhů

Nejdůležitějším z modelů je *Gauntlet*, který byl vytvořen v Blenderu. Důraz na vývoj tohoto modelu byl z důvodu, že to je klíčový prvek, který zobrazuje a předává hráči informace o elementech. Jeho hlavní částí jsou dvě postranní místa (sockety) a jeden hlavní uprostřed (zvolený element) pro krystaly, které reprezentují elementy. Dále obsahuje nespočet malých dekorativních detailů, které dohromady tvoří jeho vzhled (viz obrázek 4.14).



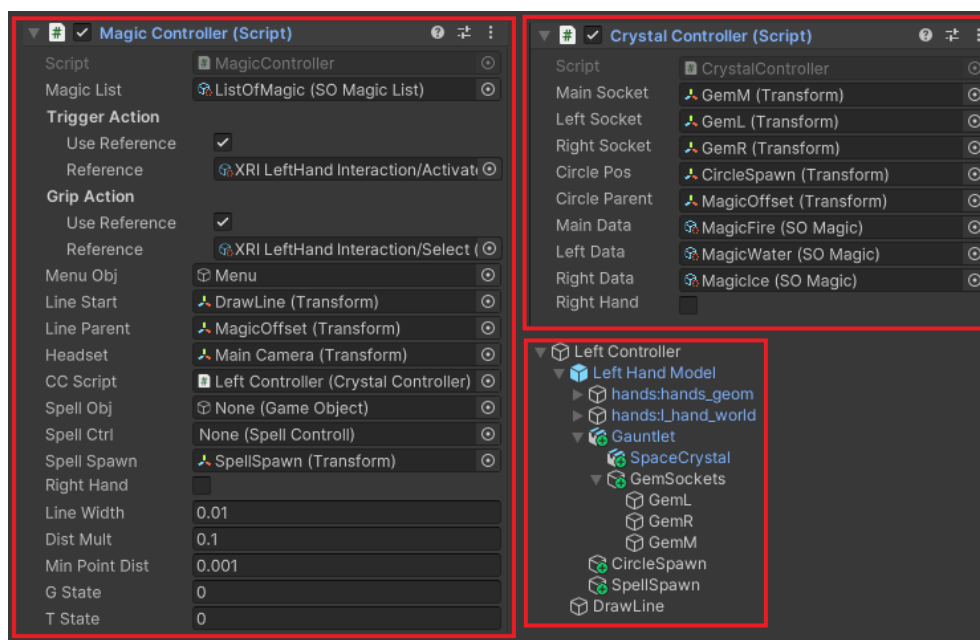
■ Obrázek 4.14 Model *Gauntlet*

4.3.3 Struktura implementace

Tato část stručně popisuje rozdělení struktury implementace kouzel a vysvětluje fungování těchto dílčích součástí. Další kapitoly tento koncept poté detailněji rozvádějí.

Celý proces tvoření kouzel je řízený dvěma scripty na každém ovladači. Jejich názvy jsou *MagicController* a *CrystalController*.

Tyto skripty obsahují všechna potřebná data a reference pro svoji funkčnost, včetně pomocné datové struktury určující konkrétní lokace z prázdných objektů (viz obrázek 4.15).



■ **Obrázek 4.15** Komponenty (Script) *MagicController* a *CrystalController*

Hlavní částí je *MagicController*, který řídí celý proces vytváření a ovládání kouzel. Jedna z jeho důležitých činností se skládá z detekce vstupů a následného zpracování stavu tlačítek Grip a Trigger (viz kapitola 4.3.6). Dále obsahuje a řídí všechny podsystémy, mezi které patří například zaznamenávání, vizualizace, zpracování a rozpoznání gest (viz kapitola 4.3.8).

CrystalController je skript, který se stará o všechny věci potřebné pro vizualizaci, prohození a získání dat z magických krystalů (viz kapitola 4.3.7)). Tento skript je ovládán voláním z *MagicController*, čímž zjednodušuje práci s výběrem a použitím elementů.

Kromě těchto řídicích skriptů jsou ještě využity *ScriptableObject* pro uložení statických dat (definice kouzel a elementů) (viz kapitola 4.3.5).

4.3.4 Popis *ScriptableObject*

ScriptableObject představuje speciální třídu v herním enginu Unity, která umožňuje optimalizaci výkonu díky úspornému ukládání dat odděleně od herní logiky (scény). Tento přístup vede k čistšímu a udržitelnějšímu kódu (eliminace zásahů do zdrojového kódu při změně).

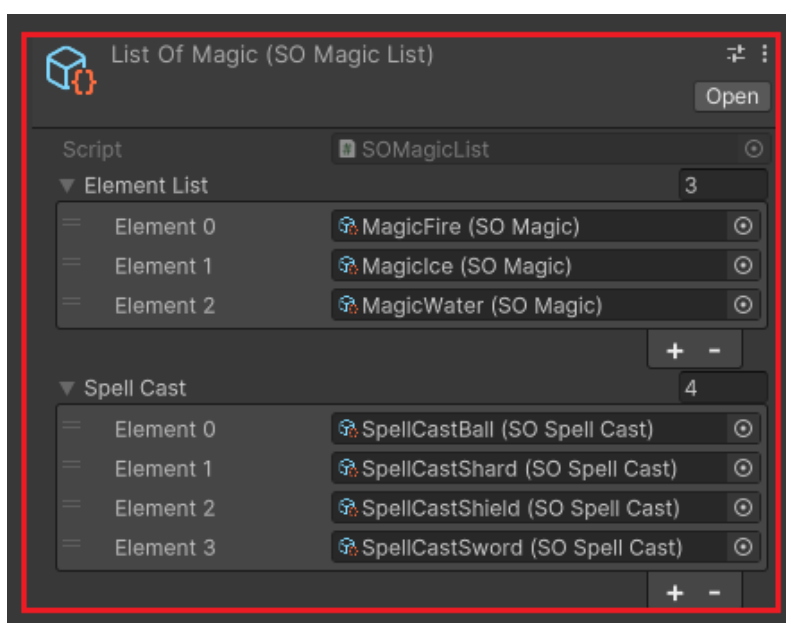
Pro použití se musí vytvořit třída odvozená od *ScriptableObject*, ve které

se specifikují požadované datové proměnné. Následně se musí vytvořit statické instance z této třídy přímo v editoru (mimo scénu). Do těchto instancí lze poté vložit data a pomocí referencí je využít ve scriptech.

4.3.5 Datová struktura pro kouzla a elementy

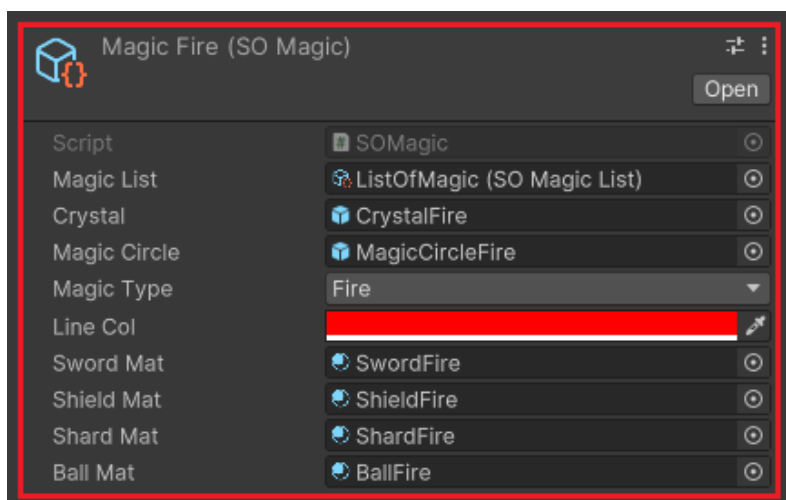
Tato část je věnována způsobu použití *ScriptableObject* pro uložení dat kouzel a elementů. Struktura dat je popsána po několika částech v následujících odstavcích.

Centrálním částí datové struktury je instance *ListOfMagic* vytvořena z *SOMagicList*. Tato instance slouží jako kolekce definovaných elementů, kouzel a enum *MagicType* s typy elementů (viz obrázek 4.16).



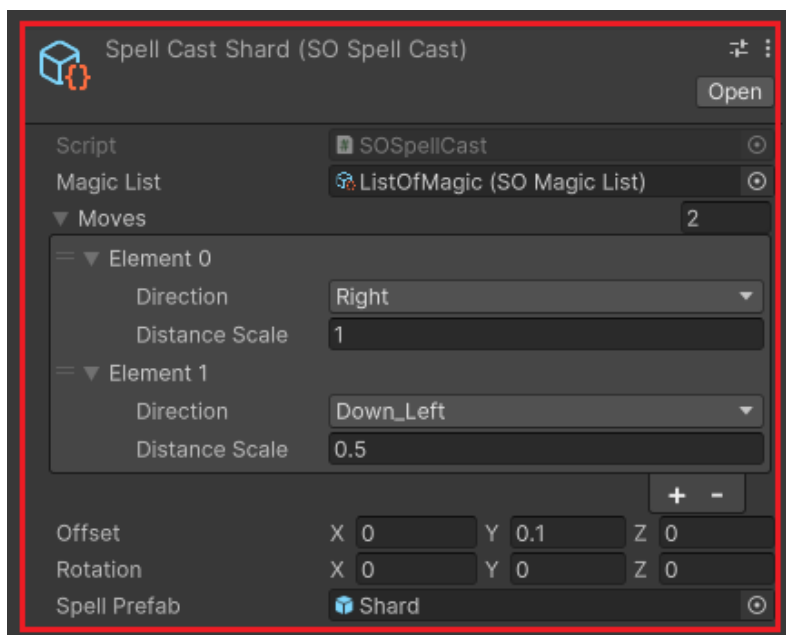
■ **Obrázek 4.16** Ukázka instance *ListOfMagic* (*SOMagicList*)

Elementy jsou definované v instancích vytvořených z *SOMagic*, které obsahují veškeré nezbytné informace. Mezi tyto informace patří prefab krystalu a magického kruhu pro vizualizaci přepínání, definice barvy používané při sesílání kouzel, označení typu elementu pomocí hodnoty enum *MagicType* a specifikace materiálů pro všechny typy kouzel (viz obrázek 4.17).



■ **Obrázek 4.17** Ukázka instance *MagicFire* (*SOMagic*)

Kouzla jsou definované v instancích vytvořených z *SOSpellCast*. Každá instance obsahuje mimo jiné reference na prefab kouzla s definicí jeho offsetu a rotace. Důležitou částí je definice pohybů potřebných pro seslání kouzla, která je strukturovaná jako kolekce dvojic složená z enum *Direction* (směrů) a proporcionálních hodnot délky (viz obrázek 4.18).



■ **Obrázek 4.18** Ukázka instance *SpellCastShard* (*SOSpellCast*)

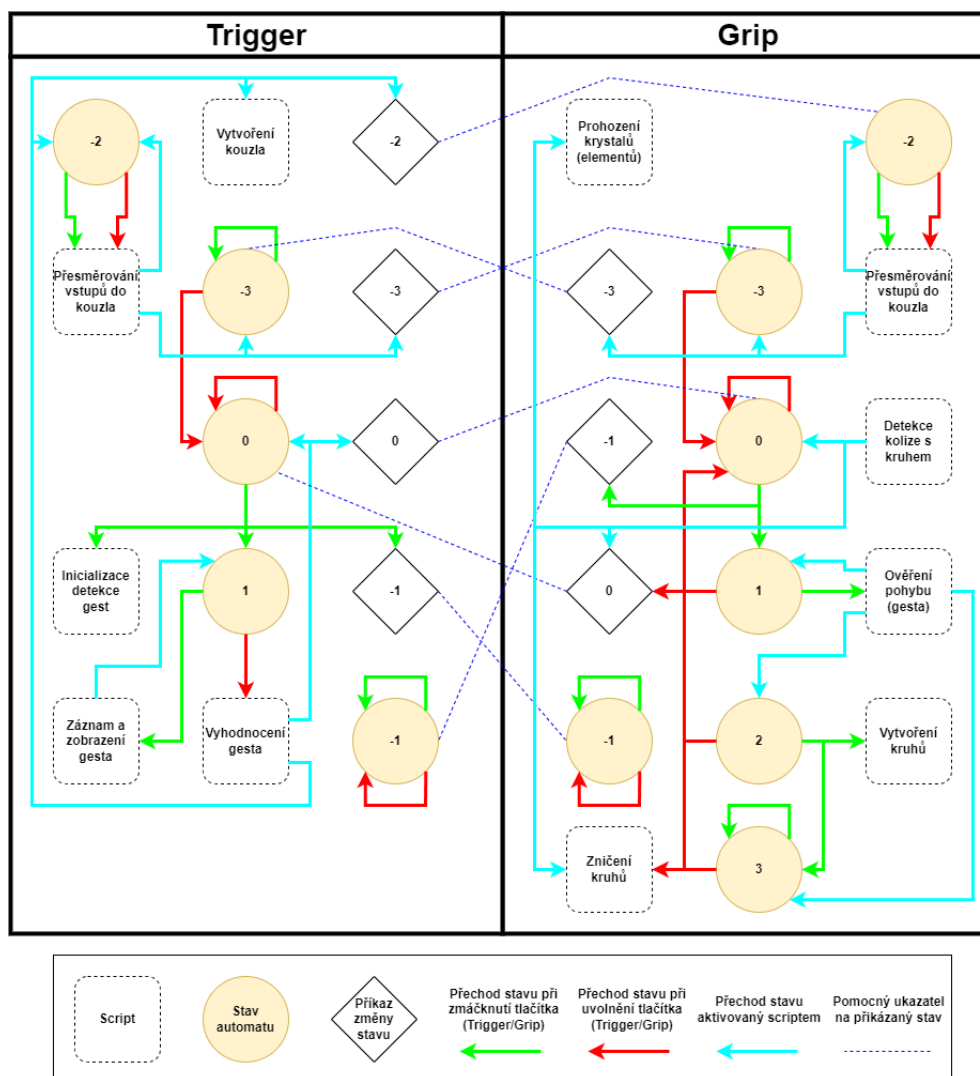
4.3.6 Zpracování vstupů

V této části je popsán postup, jak se zpracovávají vstupy z ovladače společně s hlavní rozhodovací logikou.

Celé zpracování vstupů se odehrává ve scriptu *MagicController*, kde je celá tato logika obsažena ve funkci *FixedUpdate()*. Tento název funkce má speciální význam v Unity, který způsobuje, že se automaticky spouští ve předem definovaných intervalech. Tato frekvence je defaultně nastavená na 50 volání za sekundu, ale z důvodu, že zobrazovací schopnosti zařízení ve VR začínají na 90 FPS, byla tato hodnota zvýšena na 100.

Hlavní logiku zpracování vstupů řídí dva oddělené stavové automaty, které v určitých případech mohou měnit vzájemný stav. Oba tyto automaty mají jako vstupní data svůj momentální stav a aktuální stav přiděleného tlačítka. Momentální stav těchto automatů je uložen ve dvou proměných (int) *gState* pro tlačítko Grip a *tState* pro tlačítko Trigger. Jedinou výjimkou je případ, kdy je otevřené Menu, v tomto případě se simuluje uvolnění obou tlačítek, dokud není Menu zavřeno. Změna stavu a řízení ostatních procesů probíhá vždy při spuštění funkce *FixedUpdate()*.

Obrázek 4.19 znázorňuje pomocí grafu oba výše zmíněné stavové automaty a přechody mezi jejich stavy. Podrobnější popis částí tohoto grafu je napsán níže.



■ **Obrázek 4.19** Graf stavů a přechodů pro Grip a Trigger

Levá polovina grafu obsahuje stavy pro Trigger a pravá polovina pro Grip. Tento graf se skládá ze 3 typů entit:

- **Stav automatu** – Znázorňuje momentální stav automatu ve své půlce.
- **Příkaz změny stavu** – Nastaví stav automatu v opačné půlce (z jakéhokoliv stavu).
- **Script** – Obsahuje vnější logiku (popsané v následujících kapitolách).

Mezi nimi se přechází pomocí následujících podmíněných přechodů:

- **Zelený (zmáčnutí)** – Je průchozí z aktuálního stavu, pokud je stlačeno tlačítko na dané půlce.

- **Červený (uvolnění)** – Je průchozí z aktuálního stavu, pokud je uvolněné (nestlačené) tlačítko na dané půlce.
- **Modrý (logický)** – Je spouštěn scriptem, který externě rozhoduje kdy je průchozí.

Stavy (−3) až (0) mají totožný význam pro obě půlky grafu, když se v nich automat nalézá. Zbytek stavů má odlišný význam pro každý automat, proto jsou uvedené samostatně.

- **Stav (0)** – Výchozí stav, ve kterém se automat nachází.
- **Stav (−1)** – Vstup je blokován, nemůže se z něj sám dostat.
- **Stav (−2)** – Vstup je přeposílán do seslaného kouzla.
- **Stav (−3)** – Vstup čeká na uvolnění tlačítka, po vyslání kouzla.
- **Stav (1, Trigger)** – Dokud zůstává Trigger stlačený, posílá se pozice ovladače pro záznam a vyhodnocení gesta.
- **Stav (1, Grip)** – Dokud zůstává Grip stlačený, detekuje pohyb ruky pro prohození krystalů.
- **Stav (2, Grip)** – Dokud zůstává Grip stlačený, zobrazuje magické kruhy pro prohození krystalů.
- **Stav (3, Grip)** – Po uvolnění tlačítka Grip, zničí tyto magické kruhy.

4.3.7 Prohození elementů

Proces prohození elementů lze rozdělit na následujících několik částí. Tyto části jsou níže detailněji popsány.

- Ověření pohybu (gesta)
- Vytvoření (magických) kruhů
- Detekce kolize s (magickým) kruhem
- Zničení (magických) kruhů
- Prohození krystalů

První částí je **ověření pohybu (gesta)**, která probíhá ve scriptu *Magic-Controller* (viz výpis z kódu 4.3). Nejdříve se při stisknutí tlačítka Grip uloží počáteční poloha ovladače, od které se začne testovat momentální pozice. Pokud se detekuje změna natočení ovladače o více než 60° nebo se uvolní tlačítko Grip, tak je nutné začít od začátku. V případě, že se nic z toho nestane a zároveň bude rozdíl pozice ve správném směru dostatečně velký, tak se zavolá funkce pro vytvoření magických kruhů.

```

1  // "gStart" = pozice ruky na začátku
2  // "gStartDir" = pozice ruky, mírně posunutá dopředu na
   // začátku
3  // výpočet při stavu 1
4  Vector3 move = gStart - transform.position;
5  if (Vector3.Angle(move, gStartDir) > 60){
6      if (move.magnitude > 0.05f) gState = 3;
7  } else if (move.magnitude > 0.2f)
8      gState = 2;
9  break;

```

■ **Výpis kódu 4.3** MagicController.cs, detekce gesta prohození krystalů

Části **prohození krystalů, vytvoření a zničení (magických) kruhů** jsou implementované jako funkce ve skriptu *CrystalController*. Tyto funkce jsou řízeny skriptem *MagicController*, který rozhoduje, kdy se mají spustit. Pro přehlednost je kromě těchto tří funkcí v seznamu níže popsána i funkce *ActiveGem()*, která slouží pro získání dat o zvoleném elementu (viz kapitola 4.3.11).

- **void SpawnCircles()** – Vytvoří magické kruhy na správné pozici dle prefabů uložených v pravém a levém krystalu.
- **void DestroyCircles()** – Zničí oba magické kruhy, které vytvořil.
- **void SwapGem(bool right)** – Prohodí aktivní krystal s krystalem v pravém nebo v levém socketu (dle pravdivostní hodnoty *right*).
- **SOMagic ActiveGem()** – Vrátí referenci na data elementu z aktivního krystalu.

Poslední ještě nezmíněnou částí je **detekce kolize ruky s magickým kruhem**. Kolize je provedena klasickým způsobem pomocí *CollisionBox* komponent, které jsou použité na ruce a obou magických kruzích. Na každém magickém kruhu je navíc tato kolize zachycena skriptem *MagicCircle*. Ten následně vyšle do ruky příkaz na prohození krystalů dle dat, které v něm byly uloženy při jeho vytvoření. Nakonec vyšle další příkaz pro zničení magických kruhů a přepnutí zpracování vstupů do výchozího stavu (viz kapitola 4.3.6).

4.3.8 Detekce gest

Celá detekce gest pro vytvoření kouzel se skládá se z následujících několika částí, které jsou obsažené ve skriptu *MagicController*.

■ Inicializace detekce gest

Nejdříve se spustí inicializace detekce gest, která připraví proměnné a resetuje předchozí hodnoty. Zejména vyprázdnění listu (*List<LDPoint>*), který ukládá všechny pozice z průběhu detekce, a nastaví potřebné parametry do komponenty *LineRenderer*. Tato komponenta se stará o vizualizaci uložených bodů v prostoru, které následně zobrazuje jako propojené čáry.

■ Záznam a zobrazení gest

Následně se zaznamenává pohyb ovladače do listu (*List<LDPoint>*), kde každý *LDPoint* reprezentuje jeho pozici a rotaci. Zde je nutné zmínit, že zde probíhá výpočet, který přepočítává pozici na relativní oproti počátku a zaokrouhluje je do určené mřížky. Navíc optimalizuje počet uložených pozic blokováním uložení, dokud je v určené minimální vzdálenosti od předchozí. Tyto uložené relativní pozice se následně přičtou k momentální pozici hráče a pošlou do komponenty *LineRenderer* pro vizualizaci.

■ Výpočet tvaru gest

Výpočet tvaru gest je komplexnější proces, který je podrobněji rozebrán v následující kapitole 4.3.9 společně s definicí *LDLine*. Výsledkem tohoto výpočtu je list směrových vektorů (*LDLine*) definovaných svojí délkou a hodnotou enum *Direction*, která určuje základních 6 směrů a jejich smysluplné kombinace (viz kapitola 4.3.9).

■ Nalezení kouzla s podobným tvarem

Nakonec proběhne postupné porovnání databáze kouzel (viz kapitola 4.3.5) a výše zmíněného listu *LDLine*. Následně se v každém kouzlu porovná sled směrů *Direction* s listem *LDLine*. Na první definici kouzla, která najde schodu, se zavolá vytvoření kouzla. Způsob jak probíhá vytvoření kouzla je popsán v kapitole 4.3.11.

4.3.9 Výpočet tvaru gest

Výpočet tvaru gest se skládá z procesů zredukování počtu bodů (pozic) a přepočítání této posloupnosti na směrové vektory *LDLine*.

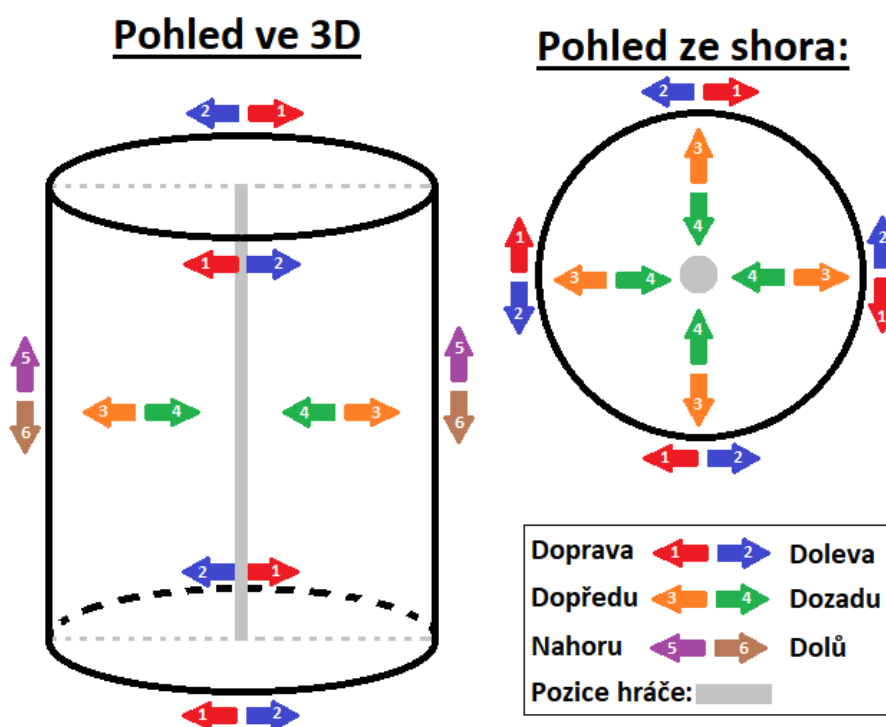
Prvním krokem je zredukování počtu bodů, které probíhá rekurzivním průchodem všech bodů a vymazáním bodů zbytečných. Zbytečný bod nepřidává žádnou důležitou informaci v zadané míře aproximace a po jeho odstranění zůstane tvar spojený ze zbývajících bodů přibližně stejný. Na jejich detekci se používá vzorec pro výpočet nejmenší vzdálenosti bodu x_0 od přímky tvořené body x_1 , x_2 (viz výpis z kódu 4.4). Tento vzorec je i s postupem podrobně vysvětlen v následující kapitole 4.3.10.

```
1 float DistPointToLine(Vector3 X0, Vector3 X1, Vector3 X2)
   { return Vector3.Cross(X2 - X1, X1 - X0).magnitude /
     (X2 - X1).magnitude; }
```

■ Výpis kódu 4.4 MagicController.cs, funkce DistPointToLine()

Po této redukci se vždy ze dvou následujících bodů a pozice headsetu vytvoří instance *LDLine*. Ta ve svém konstruktoru vypočítá relativní směr této úsečky. Tento relativní směr je spočítán speciálním způsobem kolem hráče (viz obrázek 4.20). Dle tohoto obrázku je vždy kolem hráče pomyslný válec, dle kterého se určuje směr gest následovně:

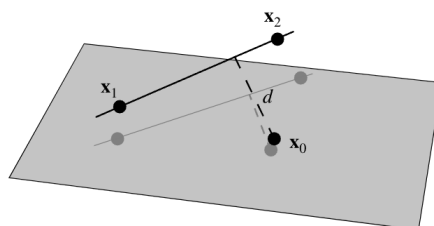
- **Dolů** – Je směrem dolů.
- **Nahoru** – Je směrem nahoru.
- **Dopředu** – Je směrem ven z válce.
- **Dozadu** – Je směrem dovnitř válce.
- **Doprava** – Je po směru hodin podél obvodu válce.
- **Doleva** – Je proti směru hodin podél obvodu válce.



■ **Obrázek 4.20** Detekce směru gesta kolem hráče

4.3.10 Vzorec nejmenší vzdálenosti bodu od přímky

Vzorec pro získání nejmenší vzdálenosti bodu od přímky (viz obrázek 4.21) pochází ze stránky Wolfram MathWorld [15]. Tato stránka je zde přepsána v českém jazyce pro vysvětlení fungování tohoto vzorce.



■ **Obrázek 4.21** Zobrazení použití vzorce [15]

Nechť je přímka v trojrozměrném prostoru určena dvěma body $\mathbf{x}_1 = (x_1, y_1, z_1)$ a $\mathbf{x}_2 = (x_2, y_2, z_2)$ ležícími na ní tak, že vektor ve směru přímky je dán vzcem

$$\mathbf{v} = \begin{bmatrix} x_1 + (x_2 - x_1)t \\ y_1 + (y_2 - y_1)t \\ z_1 + (z_2 - z_1)t \end{bmatrix}. \quad (4.1)$$

Druhá mocnina vzdálenosti mezi bodem na přímce s parametrem t a bodem $\mathbf{x}_0 = (x_0, y_0, z_0)$ je tedy

$$d^2 = [(x_1 - x_0) + (x_2 - x_1)t]^2 + [(y_1 - y_0) + (y_2 - y_1)t]^2 + [(z_1 - z_0) + (z_2 - z_1)t]^2. \quad (4.2)$$

Pro minimalizaci vzdálenosti řekneme, že $\frac{d}{dt}d^2 = 0$ a vyřešíme pro t , čímž získáme

$$t = \frac{(\mathbf{x}_2 - \mathbf{x}_1) \cdot (\mathbf{x}_0 - \mathbf{x}_1)}{|\mathbf{x}_2 - \mathbf{x}_1|^2}, \quad (4.3)$$

kde \cdot označuje skalární součin. Minimální vzdálenost lze pak nalézt dosazením t zpět do (4.2), čímž získáme

$$\begin{aligned} d^2 &= (x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2 \\ &\quad + 2t[(x_2 - x_1)(x_1 - x_0) + (y_2 - y_1)(y_1 - y_0) + (z_2 - z_1)(z_1 - z_0)] \\ &\quad + t^2[(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2] \\ &= |\mathbf{x}_1 - \mathbf{x}_0|^2 - 2 \frac{[(\mathbf{x}_1 - \mathbf{x}_0) \cdot (\mathbf{x}_2 - \mathbf{x}_1)]^2}{|\mathbf{x}_2 - \mathbf{x}_1|^2} + \frac{[(\mathbf{x}_1 - \mathbf{x}_0) \cdot (\mathbf{x}_2 - \mathbf{x}_1)]^2}{|\mathbf{x}_2 - \mathbf{x}_1|^2} \\ &= \frac{|\mathbf{x}_1 - \mathbf{x}_0|^2 |\mathbf{x}_2 - \mathbf{x}_1|^2 - [(\mathbf{x}_1 - \mathbf{x}_0) \cdot (\mathbf{x}_2 - \mathbf{x}_1)]^2}{|\mathbf{x}_2 - \mathbf{x}_1|^2}. \end{aligned} \quad (4.4)$$

Nakonec s použitím vektorového čtyřnásobného součinu

$$(\mathbf{A} \times \mathbf{B})^2 = \mathbf{A}^2 \mathbf{B}^2 - (\mathbf{A} \cdot \mathbf{B})^2, \quad (4.5)$$

kde \times označuje vektorový součin, pak dostaneme

$$d^2 = \frac{|(\mathbf{x}_2 - \mathbf{x}_1) \times (\mathbf{x}_1 - \mathbf{x}_0)|^2}{|\mathbf{x}_2 - \mathbf{x}_1|^2} \quad (4.6)$$

a odmocněním získáme krásný vzorec

$$\begin{aligned} d &= \frac{|(\mathbf{x}_2 - \mathbf{x}_1) \times (\mathbf{x}_1 - \mathbf{x}_0)|}{|\mathbf{x}_2 - \mathbf{x}_1|} \\ &= \frac{|(\mathbf{x}_0 - \mathbf{x}_1) \times (\mathbf{x}_0 - \mathbf{x}_2)|}{|\mathbf{x}_2 - \mathbf{x}_1|}. \end{aligned} \quad (4.7)$$

Zde je čitatel jednoduše dvojnásobek plochy trojúhelníku tvořeného body \mathbf{x}_0 , \mathbf{x}_1 , \mathbf{x}_2 , a jmenovatel je délka jedné ze základů trojúhelníku, což vyplývá z běžného vzorce pro plochu trojúhelníku, $\Delta = \frac{1}{2}bh$.

4.3.11 Vytvoření kouzla

Tato část popisuje proces vytvoření kouzla, poté co je gesto úspěšně rozpoznáno. Klíčové části tohoto procesu jsou ukázané ve výpisu kódu 4.5.

```

1 //vytvoreni zvoleneho kouzla
2 spellObj = Instantiate(magicList.spellCast[spell].
    spellPrefab, spellSpawn);
3 //ulozeni reference na SpellControll
4 spellCtrl = spellObj.GetComponent<SpellControll>();
5 //vlozeni informaci o zvolenem elementu
6 spellCtrl.SetData(cCScript.ActiveGem(), this);

```

■ Výpis kódu 4.5 MagicController.cs, vytvoření kouzla

Při úspěšném rozpoznání gesta se uloží do proměnné *spell* číslo pořadí (index) patřičného kouzla, které se následně použije pro získání prefabu z *ListOfMagic* (viz kapitola 4.3.5). Z tohoto prefabu se poté vytvoří instance kouzla (vznikne ve scéně) na pozici uložené v *spellSpawn*. Reference na tuto instanci se poté uloží do proměnné *spellObj* pro další použití.

Následně se uloží reference na script *SpellControll* z instance *spellObj*. Každý prefab kouzla je ovládán skriptem, který dědí ze skriptu *SpellControll*. To umožňuje definovat stejné funkce pro práci s každým kouzlem (interface). Tento přístup poté usnadňuje práci s kouzly, nejen pro přeposílání vstupů (viz kapitola 4.4).

Nakonec se vloží data z momentálně vybraného elementu a referenci na script *MagicController* pomocí funkce *spellCtrl.SetData()*. Tato funkce nastaví všechny náležitosti, které jsou potřeba pro vytvoření a následné použití kouzla.

4.4 F6: Použití kouzel

V této kapitole je nejdříve popsáno chování jednotlivých kouzel v různých situacích a poté způsob, jakým je toho dosaženo.

4.4.1 Chování kouzel

V této části je popsáno chování jednotlivých kouzel. Všechny kouzla mají zapnuté kolize s dalšími objekty. Existují speciální objekty reagující na kouzla (viz kapitola 4.5). Tyto interakce s objekty probíhají při detekci kolize a mají ji pouze některá kouzla. Pro přehlednost jsou základní vlastnosti kouzel popsány v tabulce 4.1, která říká co se stane při kolizi a jestli interagují s objekty.

Kouzlo	Při kolizi	Interakce při kolizi
koule	zmizí	ano
krystal	zasekne se a po 5 sekundách zmizí	ano
meč	–	ano
štít	sníží se výdrž o 1, pokud se dostane na 0, zmizí	ne

■ **Tabulka 4.1** Vlastnosti kouzel

Chování těchto kouzel na vstupy ovladačů (Grip/Trigger) jsou popsány v následujícím seznamu:

- **Koule** – Stlačením a následným uvolněním Triggeru se zahodí.
- **Krystal** – Po stlačení Triggeru vystřelí dopředu, dle své orientace.
- **Meč** – Po stlačení Triggeru zmizí.
- **Štít** – Po stlačení Triggeru zmizí.

4.4.2 Princip řízení kouzel

V této části je vysvětleno, jakým způsobem je docíleno výše specifikované chování kouzel (viz kapitola 4.4.1). Aby fungovalo chování kouzel, musí každý prefab obsahovat potomka scriptu *SpellControll*. Interface tohoto scriptu totiž obsahuje 3 veřejné funkce a jednu funkci detekující kolize, kterými se kouzlo nastavuje a ovládá. Všechny tyto funkce nejdříve zpracují vstupní parametry a poté zavolají jejich přidruženou abstraktní funkci (implementovaná v potomkovi).

- **void SetData(SOMagic magicType, MagicController magicController)** – Uloží referenci na *SOMagic* (element) a *MagicController*. Poté spustí abstraktní funkci *SetMaterial()*, která nastaví správný materiál pro dané kouzlo.

- **bool Trigger(bool pressed)** – Je neustále volána s aktuálním stavem stlačení Triggeru. Pokud nedojde ke změně oproti předchozímu volání, vrací se hodnota false (nepravda). V opačném případě se přeepoše nový stav do abstraktní funkce *bool InputTrigger(bool pressed)* a vrátí se její zpětná hodnota.
- **bool Grip(bool pressed)** – Je neustále volána s aktuálním stavem stlačení Gripu. Pokud nedojde ke změně oproti předchozímu volání, vrací se hodnota false (nepravda). V opačném případě se přeepoše nový stav do abstraktní funkce *bool InputGrip(bool pressed)* a vrátí se její zpětná hodnota.
- **void OnCollisionEnter(Collision other)** – Při detekci začátku kolize přeepoše proměnnou *other* do abstraktní funkce *MyColl(Collision other)*. Poté se pokusí získat z *other* objekt a v něm komponentu *SpellInteract*, pokud se povede nalézt, tak se na ní zavolá funkce *SpellContact(magicType)* (viz kapitola 4.5). Tato funkce má pouze ve štítu vypnuté volání *SpellContact(magicType)*.

4.5 F7: Interakce kouzel s okolím

V této kapitole je nejdříve popsán princip systému pro interakce kouzel s okolím a poté jsou zde popsány dva příklady těchto interakcí.

Každý objekt, který má speciálně reagovat na kontakt s kouzlem, musí obsahovat potomka abstraktního scriptu *SpellInteract*. V případě, že ho neobsahuje, tak se neprovede žádný speciální efekt kouzla, ale fyzikálně na sebe stále normálně reagují. Jediným účelem scriptu *SpellInteract* je umožnění volání funkce *void SpellContact(SOMagic magicType)* v jakémkoliv potomkovi. Tato funkce je volána při kolizi s jakýmkoliv kouzlem kromě štítu (viz kapitola 4.4.2). Parametr v ní obsahuje typ elementu kouzla, se kterým byla detekována kolize.

Pro ukázkou tohoto systému jsou vytvořené dva typy objektů s různým chováním interakcí. Scripty, které jsou níže zmíněné jsou potomky *SpellInteract*.

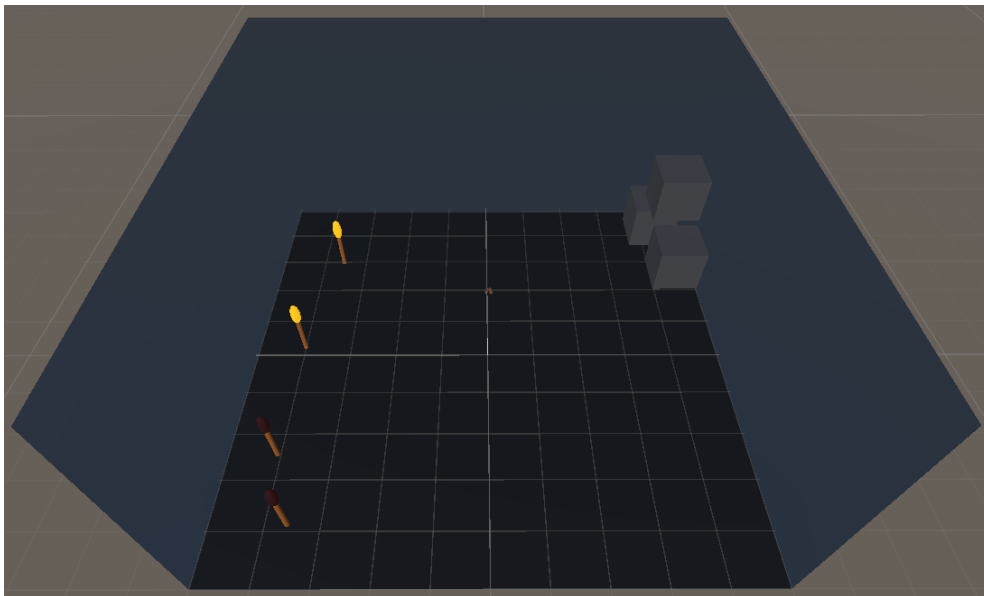
- **Pochodeň** – Má na sobě script *SpellInteractTorch*, který přepíná pochodeň mezi zapáleným a uhaslým stavem. Zapálí se ohnivým a uhasí se ledovým nebo vodním elementem kouzla.
- **Zničitelná krabice** – Má na sobě script *SpellInteractDestroy*, který zničí tento objekt jakýmkoliv elementem kouzla.

Popis navrženého řešení

V této kapitole je stručně popsán způsob ovládání, vzhled a možnosti navrženého řešení.

5.1 Popis prostředí (scény)

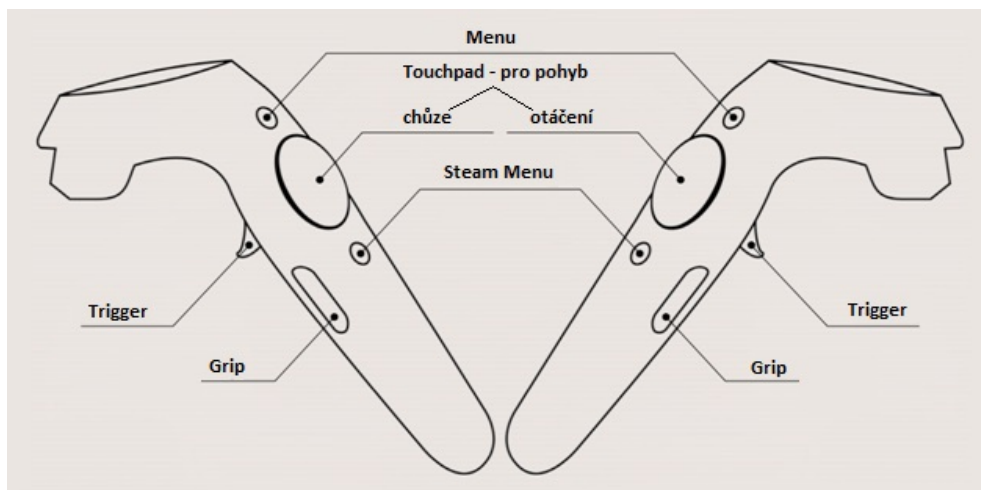
Celé prostředí se nachází v uzavřené místnosti, po které se může hráč pohybovat (viz obrázek 5.1). Tato místnost obsahuje hráče (jdou vidět pouze rukavice), 4 zapalovatelné pochodně a 3 zničitelné krabice (pomocí kouzel).



■ Obrázek 5.1 Místnost s objekty

5.2 Ovládání

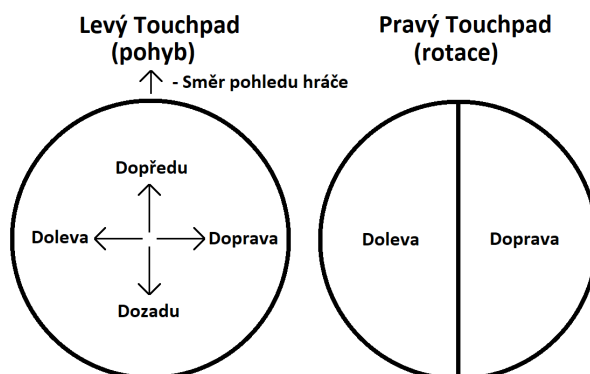
Ovládání lze rozdělit na několik částí, které jsou jednotlivě popsány v následujících podkapitolách. Mezi tyto části patří: pohyb hráče, ovládání menu a gesta pro kouzlení. Na následujícím obrázku 5.2 jsou popsána tlačítka, která se používají pro tyto činnosti.



■ Obrázek 5.2 Rozvržení ovládání v nárzeném řešení

5.2.1 Pohyb hráče

Pohyb hráče ve virtuálním prostředí lze rozdělit na dva hlavní způsoby. První využívá fyzický pohyb, který je přenášen přímo do hry. Druhá metoda spočívá ve využití Touchpadů na ovladačích, které simulují chůzi a rotaci hráče. Schematické rozložení ovládacích funkcí pro pohyb na Touchpadech je k dispozici na obrázku 5.3.



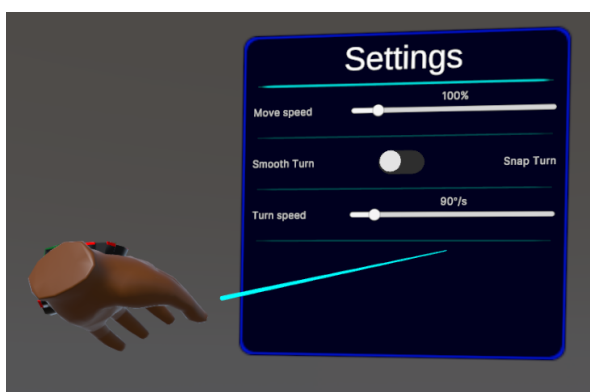
■ Obrázek 5.3 Schématické rozložení Touchpadů pro pohyb

5.2.2 Ovládání menu

V této části je vysvětleno ovládání menu.

Pro otevření/skrytí je potřeba stlačit tlačítko Menu. Po dobu, kdy je menu aktivní, je ovládání gest pro kouzlení deaktivováno.

Po aktivaci menu se při zaměření ovladače na libovolný jeho segment objeví vizuální indikátor ve formě čáry. Tento indikátor znázorňuje bod, na který ovladač v menu míří (viz obrázek 5.4). Veškeré interaktivní prvky menu lze intuitivně ovládat stisknutím tlačítka Trigger na ovladači.



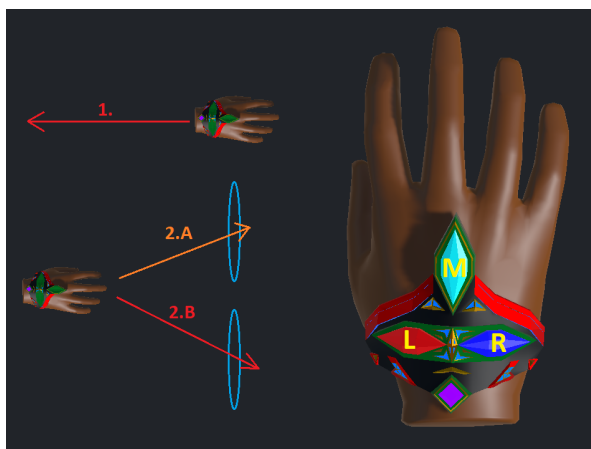
■ Obrázek 5.4 Menu s indikátorem míření

5.2.3 Gesta pro kouzlení

V této části popsány způsoby vytváření kouzel.

Způsob použití těchto kouzel je vysvětlený v kapitole 5.3 níže. Tvoření kouzel se skládá ze dvou částí, výběr elementu (led/ohněň/voda) a zvolení typu (meč/štít/koule/krystal).

Hráč má na každé ruce 3 krystaly definující elementy, ze kterých je vždy jeden vybraný (prostřední). Pro změnu vybraného krystalu je potřeba provést na dané ruce gesto pro změnu elementu. Toto gesto se skládá z kroků ukázaných na obrázku 5.5.

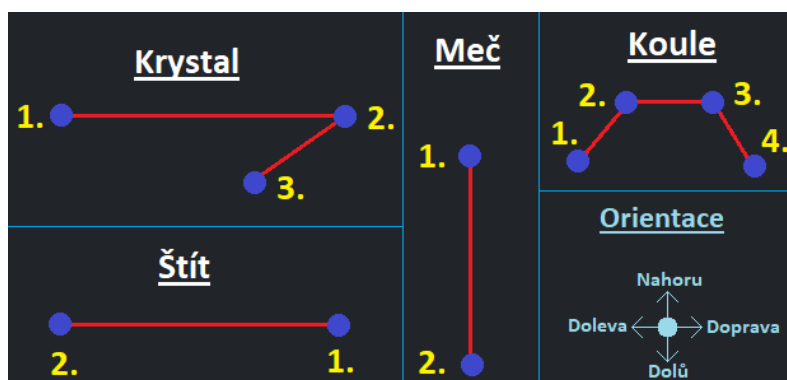


■ **Obrázek 5.5** Rozmístění krystalů a gesto pro jejich prohození

Výše zobrazený průběh gesta pro prohození krystalu probíhá následujícím způsobem:

1. Začíná stlačením a držetím Gripu.
2. Pokračuje posunutím ruky v přesně opačném směru, než ukazuje prostředníček, dokud se neobjeví 2 magické kruhy (viz obrázek 4.13).
3. Dokončí se prostrčením ruky skrz magický kruh s barvou zvoleného elementu (2.A -> L / 2.B -> R).

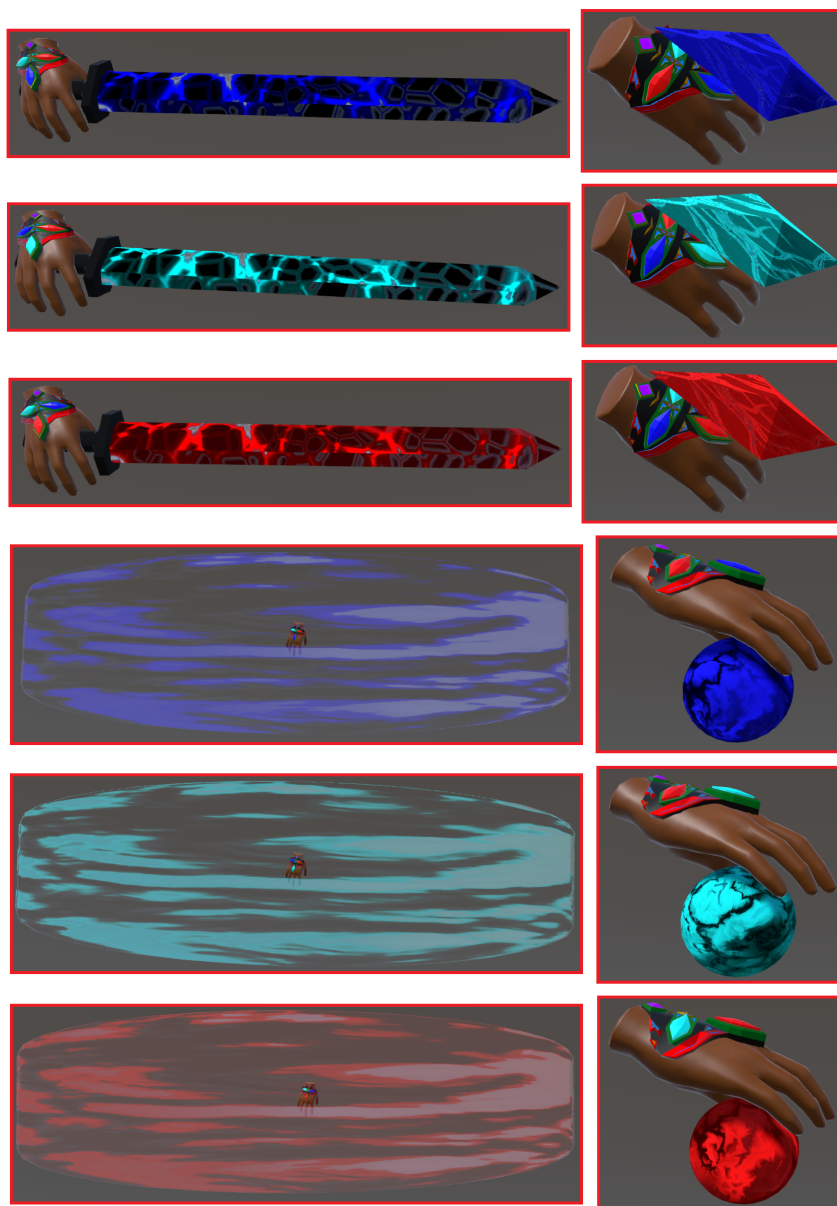
Pro zvolení typu kouzla se musí nakreslit správný tvar. Kreslení se spustí stlačením Triggeru. Po uvolnění Triggeru se tento tvar vyhodnotí a pokud je rozeznán korektní tvar, tak se kouzlo vytvoří ve vybraném elementu. V obrázku 5.6 jsou korektní tvary, které specifikují typ kouzla. Orientace směru se řídí dle obrázku 4.20 v kapitole (viz kapitola 4.3.1).



■ **Obrázek 5.6** Tvary gest kouzel

5.3 Kouzla

V této části je popsáno použití a vzhled všech implementovaných kouzel. Existují 4 typy kouzel (Koule, Krystal, Meč, Štít), které je možné vytvořit v každém ze 3 elementů. Celkový počet kouzel je 12 (viz obrázek 5.7).



■ Obrázek 5.7 Ukázka vytvořených kouzel

Ovládání těchto kouzel se dělí dle typu. Způsob ovládání každého typu kouzla a jeho použití je vysvětleno v tabulce 5.1.

Kouzlo	Ovládání	Použití
koule	zahodí se stlačením a uvolněním Triggeru	útok na dálku
krystal	stlačením Triggeru vystřelí dopředu, dle své orientace	útok na dálku
meč	stlačením Triggeru zmizí	útok na blízko
štít	stlačením Triggeru zmizí	obrana

■ **Tabulka 5.1** Použití kouzel

5.4 Interaktivní objekty

V této části jsou posány všechny objekty, které reagují element (led/ohně/voda) útočného kouzla při kontaktu. Všechny takto interaktivní objekty jsou vyjmenovány v následujícím seznamu společně s vysvětlením jejich reakcí na různé elementy.

- **Pochodeň** – Zapálí se ohnivým a uhasí se ledovým nebo vodním elementem.
- **Zničitelná krabice** – Tento objekt se zničí při kontaktu s jakýmkoliv elementem.



Kapitola 6

Testování

Tato kapitola se věnuje testování a porovnání navrženého řešení a 4 analyzovaných her s následným vyhodnocením výsledků.

Testování je klíčovým prvkem ve vývoji softwaru, zejména v oblasti VR, kde uživatelská zkušenost hraje zásadní roli. Cílem tohoto testování je odhalit případné chyby a nedostatky, a zhodnotit celkovou efektivitu a použitelnost navrženého řešení.

Struktura této kapitoly je rozdělena do tří klíčových částí: metodiky testování, kvantitativní analýzy a kvalitativní analýzy.

V první části jsou detailně popsány použité postupy a specifikace testovací sestavy.

Další částí je kvantitativní analýza, která se zaměřuje na měřitelné metriky, jako je přesnost rozpoznávání gest a výkonnost systému.

Poslední část je věnována kvalitativní analýze testování, která zkoumá informace z vyplněných dotazníků účastníků a subjektivních hodnocení funkcionalit dle autora.

6.1 Metodika testování

Testování bylo provedeno na vzorku 7 uživatelů (dále zmiňováni jako účastníci) s různou úrovní zkušeností s VR technologiemi. Každý účastník strávil minimálně 20 minut procházením klíčových částí 4 analyzovaných her a navrženého řešení. Během testování byly shromažďovány údaje o spolehlivosti identifikace gest a další relevantní data pro následnou kvantitativní a kvalitativní analýzu.

Po testování každé hry účastníci vyplnili strukturovaný dotazník (viz kapitola 6.1.1), který byl navržen tak, aby zachytil jejich bezprostřední dojmy a připomínky.

Testovací sestava měla následující parametry:

- CPU: Intel Core i7-9700K
- GPU: NVIDIA GeForce RTX 2070
- RAM: 32 GB DDR4
- Systém VR: HTC Vive

6.1.1 Struktura dotazníku

Dotazník předložený účastníkům po testování každé hry obsahoval následující otázky:

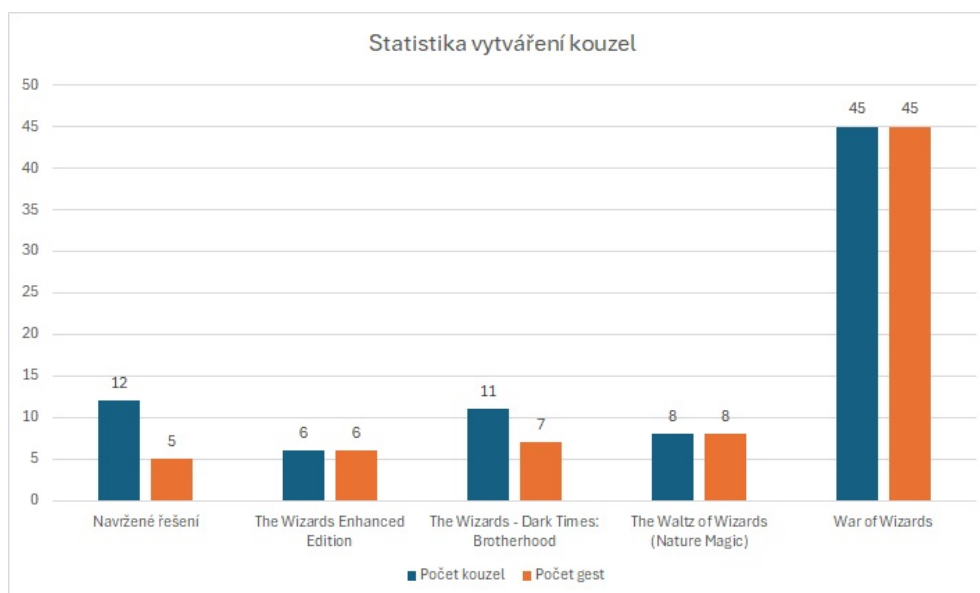
- Co vás nejvíce zaujalo?
- Byl při hraní ukázky nějaký problém?
- Co by jste vylepšil(a)?

6.2 Kvantitativní analýza

V této kapitole jsou porovnány vybrané parametry 4 analyzovaných her a navrženého řešení. Mezi tyto vybrané parametry patří poměr dostupných kouzel k počtu gest, přesnost rozpoznávání gest a výkonová náročnost systému. Výsledky jsou zobrazeny pomocí grafů. Každý graf je doprovázen komentářem vysvětlujícím, jaké závěry lze z prezentovaných dat vyvodit.

6.2.1 Statistika tvorby kouzel

Tato podkapitola prezentuje porovnání dvou zásadních metrik: množství dostupných kouzel a počet gest nezbytných pro jejich tvorbu. Data byla získána při praktickém procházení zkoumaných her.



■ **Obrázek 6.1** Graf statistiky tvorby kouzel

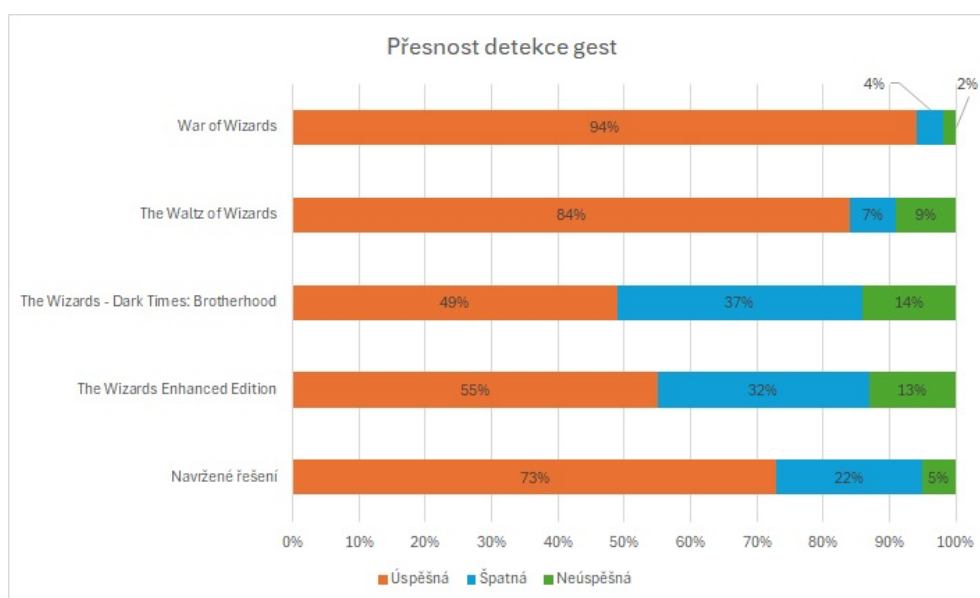
Tyto statistické údaje jasně dokazují výhody navrženého řešení. Toto řešení s pomocí 5 gest umožňuje vytvořit až 12 kouzel, čímž výrazně předčí konkurenční tituly jako The Wizards Enhanced Edition (6:6) nebo The Waltz of Wizards (8:8).

Tento efektivní design nejen usnadňuje ovládání hry, ale také otevírá prostor pro komplexnější herní scénáře bez zvyšování složitosti ovládání.

6.2.2 Spolehlivost identifikace gest

Tato podkapitola prezentuje porovnání spolehlivosti identifikace gest. Data pro tuto analýzu byla získána při praktickém testování. Každý z účastníků tohoto testování provedl minimálně 100 gest v každé ze 4 zkoumaných her a navrženého řešení. Výsledky spolehlivosti identifikace gest byly rozděleny do následující 3 kategorií:

- **Úspěšná** – Gesta, která byla správně rozpoznána.
- **Neúspěšná** – Gesta, která nebyla rozpoznána vůbec.
- **Špatná** – Gesta, která byla rozpoznána, ale jako jiné, než bylo cílem.



■ **Obrázek 6.2** Graf přesností detekce

Výsledky ukazují, že navrženého řešení dosahuje cílené úspěšnosti detekce gest, která splňuje minimální limit z nefunkčního požadavku N2 (viz kapitola 3.3). S úspěšností 73 % a pouze s 5 % neúspěšných detekcí předčí polovinu testovaných her.

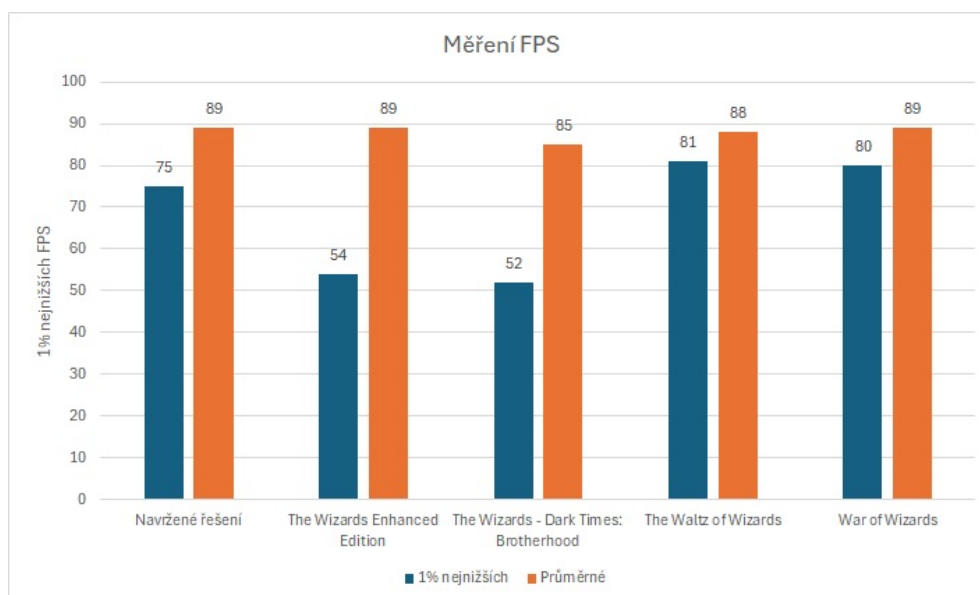
Tato vyšší přesnost detekce přispívá k plynulejšímu hernímu zážitku a snižuje frustraci hráčů při ovládání kouzel.

6.2.3 Náročnost na výkon

V podkapitole jsou porovnány výsledky výkonnostního měření 4 analyzovaných her a navrženého řešení. Data pro tuto analýzu byla naměřena programem *MSI Afterburner* [16], který se používá pro měření výkonu ve hrách.

Pro porovnání výkonu se běžně používá hodnota FPS, která udává počet snímků za sekundu. K porovnání jsou použity následující metriky, které vycházejí z hodnoty FPS: V následujícím grafu jsou použity následující dvě metriky:

- **1 % minimální FPS (1 % Low)** – Udává průměr vypočítaný z 1 % nejnižších zaznamenaných FPS.
- **Průměrné FPS (Avg FPS)** – Udává celkový průměr zaznamenaných FPS.



■ **Obrázek 6.3** Graf s porovnáním FPS

Analýzou získaných dat bylo zjištěno, že všechny testované hry dosahují velmi podobných hodnot. Je třeba poznamenat, že výsledky byly ovlivněny fyzickým limitem 90 FPS v systému VR. Jediný významnější rozdíl je patrný pouze v hodnotě 1 % minimálních FPS, který indikuje výkon systému v nejnáročnějších situacích. Na základě těchto výsledků lze konstatovat, že navržené řešení nemá zjevné výkonnostní problémy oproti konkurenci.

6.3 Kvalitativní analýza

Tato kapitola se zaměřuje na hodnocení navrženého řešení, které vychází ze zpětné vazby účastníků testování a odborného názoru autora.

Nejdříve jsou uvedeny silné stránky tohoto řešení a poté oblasti pro zlepšení.

6.3.1 Silné stránky navrženého řešení

Tato podkapitola popisuje hlavní silné stránky navrženého řešení, které byly identifikovány během testování.

1. **Inovativní kombinace kouzel a elementů** – Systém umožňuje kombinovat různé typy kouzel s elementy, což vede k velkému množství možných kouzel při zachování jednoduchosti učení. Toto řešení bylo účastníky hodnoceno velmi pozitivně.
2. **Funkční herní prostředí** – Podařilo se vytvořit plně funkční VR prostředí, ve kterém lze sesílat kouzla s očekávanými efekty. Účastníci ocenili plynulost a responsivnost systému.
3. **Flexibilní objektově orientovaný návrh** – Díky dobře navrženému OO designu je systém snadno rozšiřitelný o nové herní prvky, kouzla či elementy. To umožňuje snadný budoucí vývoj a úpravy.
4. **Intuitivní uživatelské rozhraní** – Přehledné menu nastavení a jednoduchý design byl účastníky hodnocen kladně. Většina testujících se v systému rychle zorientovala.
5. **Rychlé přepínání mezi elementy** – Systém prohazování krystalů pro změnu elementu byl hodnocen jako rychlý a intuitivní. Účastníci ocenili plynulost této akce.
6. **Vizuální zpětná vazba při sesílání kouzel** – Účastníci pozitivně hodnotili vizualizaci pohybu při sesílání kouzel, která jim pomáhala lépe pochopit a zapamatovat si jednotlivá gesta.

6.3.2 Oblasti pro zlepšení

Tato podkapitola identifikuje a popisuje části navrženého řešení, které byly na základě zpětné vazby účastníků často zmiňovány, jako potenciální oblasti pro další vylepšení.

1. **Grafické zpracování** – Účastníci často komentovali, že grafika modelů je pouze dostatečná.
 - **Nepodstatné** – Z důvodu, že zaměření této práce není vytvoření kvalitních grafických prvků a prototyp není tímto problémem nijak ovlivněn, je tento nedostatek nepodstatný.

- 2. Omezený počet interaktivních předmětů** – Účastníci by uvítali více typů předmětů, které by reagovali na kouzla.
 - **Vyřešeno** – Navržené řešení umožňuje jednoduché přidání těchto předmětů, které nebyly přidány z důvodu, že se jedná pouze o prototyp.
- 3. Přesnost detekce gest** – Někteří účastníci měli občas problémy s přesností rozpoznání gest.
 - **Vyřešeno** – Vzhledem k výsledkům kvantitativní analýzy (viz obrázek 6.2) a komplexitě problému je momentální řešení brané za dostatečné.



Kapitola 7

Závěr

Vývoj mechaniky pro tvoření kouzel pomocí gest ve virtuální realitě se podařilo dokončit včetně spustitelného prototypu. Hlavním přínosem této práce je inovativní přístup k problematice tvoření kouzel ve VR, který vychází z důkladné analýzy existujících řešení a snahy překonat jejich nedostatky.

Řešení je unikátní tím, že rozděluje tvoření kouzel na dvě dílčí části: výběr elementu a typu kouzla. Díky tomuto rozdělení lze vytvořit jakýkoliv typ kouzla za pomoci kteréhokoliv z elementů, což efektivně zvyšuje počet použitelných kouzel a zároveň nenavýšuje celkový počet gest.

Vytvořený prototyp byl důkladně zdokumentován a podroben systematickému testování. Výsledky testů prokázaly funkčnost implementovaného řešení a potvrdily značný potenciál zvoleného přístupu pro další rozvoj.

Navazující práce by mohly zahrnovat například implementaci komplexnějších herních prvků, jako jsou hádanky, které vyžadují kombinaci různých kouzel pro své řešení.

V průběhu realizace této bakalářské práce autor získal cenné zkušenosti z oblasti vývoje her.

Bibliografie

1. EDUCATION, VR. *Virtuální realita – historie a současnost* [online]. [cit. 2024-03-15]. Dostupné z: <https://vreducation.cz/virtualni-realita-historie-a-soucasnost/>.
2. KHRONOS. *OpenXR* [online]. [cit. 2024-03-15]. Dostupné z: <https://www.khronos.org/openxr/>.
3. NEPOR, Vladimír. *What XR controllers exist & why we need Input Manager* [online]. GameArter, 2020 [cit. 2024-02-15]. Dostupné z: <https://www.gamearter.com/blog/xr-input-manager-controllers>.
4. STUDIO, Carbon. *The Wizards - Enchanced Edition* [online]. Carbon Studio, 2018 [cit. 2024-02-15]. Dostupné z: https://store.steampowered.com/app/586950/The_Wizards__Enhanced_Edition/.
5. GALYONKIN, Sergey. *SteamSpy* [online]. [cit. 2024-02-15]. Dostupné z: <https://steamspy.com/>.
6. STUDIO, Carbon. *The Wizards - Dark Times: Brotherhood* [online]. Vertigo Games, 2020 [cit. 2024-02-15]. Dostupné z: https://store.steampowered.com/app/1103860/The_Wizards__Dark_Times_Brotherhood/.
7. DYNAMICS, Aldin. *Waltz of the Wizard* [online]. Aldin Dynamics, 2019 [cit. 2024-02-15]. Dostupné z: https://store.steampowered.com/app/1094390/Waltz_of_the_Wizard/.
8. ENTERTAINMENT, Arcane Miracle. *War of Wizards* [online]. Arcane Miracle Entertainment, 2022 [cit. 2024-02-15]. Dostupné z: https://store.steampowered.com/app/2009460/War_of_Wizards/.
9. TECHNOLOGIES, Unity. *Unity* [online]. [cit. 2024-04-05]. Dostupné z: <https://unity.com/products/unity-engine>.
10. TECHNOLOGIES, Unity. *Unity Asset Store* [online]. [cit. 2024-04-05]. Dostupné z: <https://assetstore.unity.com/>.
11. FOUNDATION, Blender. *Blender* [online]. [cit. 2024-04-15]. Dostupné z: <https://store.steampowered.com/app/365670/Blender/>.

12. TUTORIALS, Valem. *Oculus Hands How to Make a VR Game 2022* [online]. [cit. 2024-04-15]. Dostupné z: <https://drive.google.com/file/d/10b39IekUdpBHlcTslZ-B1NRyH5uqPUe1/view>.
13. DYMENTAL. *34 Free Shader FX Masks* [online]. [cit. 2024-04-15]. Dostupné z: <https://assetstore.unity.com/packages/2d/textures-materials/34-free-shader-fx-masks-71311>.
14. QUEST5. *Magic Circle* [online]. [cit. 2024-04-15]. Dostupné z: <https://forum.arcaneodyssey.dev/t/magic-circles-public-use/30510>.
15. RESEARCH, Wolfram. *Point-Line Distance-3-Dimensional* [online]. [cit. 2024-04-15]. Dostupné z: <https://mathworld.wolfram.com/Point-LineDistance3-Dimensional.html>.
16. MSI. *MSI Afterburner* [online]. [cit. 2024-04-15]. Dostupné z: <https://www.msi.com/Landing/afterburner/graphics-cards>.

Obsah příloh

build	složka se spustitelnou formou implementace
src	
├── impl	zdrojové kódy implementace
└── thesis	zdrojová forma práce ve formátu \LaTeX
text	text práce
└── thesis.pdf	text práce ve formátu PDF