FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

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 srovnejte s výsledky analýzy provedené v bodu 1.
8. Vytvořte dostatečnou dokumentaci.

Elektronicky schválil/a Ing. Michal Valenta, Ph.D. dne 22. prosince 2023 v Praze.

Bakalářská práce

IDENTIFIKACE GEST A PROVÁDĚNÍ KOUZEL VE VIRTUÁLNÍ REALITĚ

Jakub Vosička

Fakulta informačních technologií
Katedra softwarérového inženýrství
Vedoucí: doc. Ing. Mgr. Petr Klán, CSc.
19. května 2024

Obsah

Seznam obrázků

Seznam tabulek

Seznam výpisů kódu

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 16. května 2024

Abstrakt

Cílem bakalářské práce je analizovat již existujících VR hry, s cílem navrhnout a implementovat mechaniky pro sesílání alespoň 3 kouzel v herním prostředí VR. Práce analizuje 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, využívají 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 zkratek

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 Untiy
crafting	proces vytváření ze základních předmětů, předměty jiné
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
iterace	opakovací cyklus (2 iterace = 2 opakování)
HTC Vive	jeden ze systému pro používání VR
mana	magická energie
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 sesílání kouzel pomocí gest, společně s testovacím prostředím ve kterém je možné si kouzla vyzkoušet. Návrh hlavní mechaniky by se měl zakládat na výsledku podrobné analýza vybraných VR her s porovnání kladů a záporů jejich přístupů. Analýza také prozkoumává problémy, které s vývojem mechanik pro sesílání kouzel souvisí a jakým způsobem k nim přistupují zkoumané hry. Z výsledků této analýzy následně navrhnout nový inovativnější návrh, který bude přistupovat na problematiku novým způsobem. Finální produkt by měl obsahovat možnost sesílat minimálně 3 kouzla.

Úvod

Virtuální realita se stává stále významnější v 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 v způsobu prožívání zážitků v herním prostředí, který otevírá také dveře k 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 ovládání, 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 scifi či fantasy, zároveň kvůli rychlému růstu v technologiích VR, je zde stále nevyužitý prostor pro vytváření nových herních zážitků, jelikož trh ještě nebyl zcela zahlcen. Proto si práce klade za cíl zkoumat, navrhnout a implementovat inovativní herní mechanismy pro sesílání kouzel pomocí gest ve virtuální realitě. Vyvinutí této technologie přináší nové možnosti, jak obohatit herní zážitek uživatelů a posunout hranice toho, co je v rámci VR možné.

Bakalářská práce obsahuje analýzu 4 vybraných her, u kterých zhodnotí jejich přístup k sesílání kouzel. Pomocí výsledků analýzy a názorů autora poté bude navrhnut nový návrh, jak sesílat a ovládat kouzla, který bude v praktické části realizován v herním enginu Unity společně s řešením problémů vyskytnutých při implementaci návrhu.

Kapitola 1
Analýza

1.1 Základní informace o VR

Technologie VR je tu s námi 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 samostatný (bez PC) nebo ho připojíte k hardwarově silnému PC nebo herní konzoly jako PlayStation 4. Cílem virtuální reality je prezentovat fiktivní digitální svět v pokročilé simulaci včetně snímání vašeho pohybu. Umožní vám nahlédnout do vzdálených míst naší planety, vyzkoušíte si extrémní zážitky nebo si zahrajete opravdu intenzivní hry a nakonec oddychnete u prostorových filmů. V dnešní době je virtuální realita jakožto technologie zcela někde jinde, například se v lékařství VR používá na léčbu nemocí, poruch, dokonce i ztráty paměti. Dalším využití je pomocí 3D modelů pro výuku anatomie jako nikdy předtím nebo plánování operací a nácvik operací pro chirurgy. Sportovní trénink ve VR umožňuje replikovat dokonalé podmínky, jako při reálném výkonu činnosti. VR se pro vzdělávání nepoužívá jen na univerzitách a školách, ale i ve firmách. Díky simulacím situací ve virtuální realitě, které by jinak byly těžce 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. Většina lidí si i dnes pod pojmem VR vybaví nějakou hru než spíše firemní a korporační využití. Díky VR je dnes možný meeting ve virtuální místnosti s každým účastníkem fyzicky v jiné části 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 specifikaci, kterou mohou

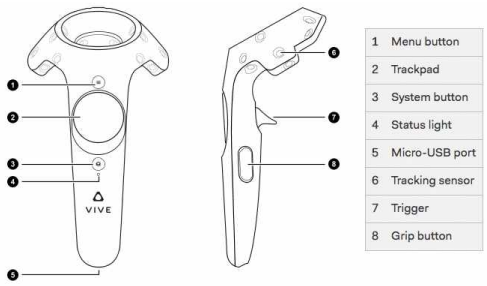
dodržovat, aby zajistili kompatibilitu svého systému s minulými, současnými a budoucími XR aplikacemi. Vývojáři aplikací se již nemusí starat o cílové platformy, protože OpenXR zajišťuje, že aplikace se bude chovat podobně na všech konformních zařízeních. Spotřebitelé si nyní mohou vybrat svůj preferovaný headset místo toho, aby se rozhodovali na základě kompatibility se svou oblíbenou aplikací. OpenXR standardizuje použití škály schopností přesplatformních XR 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

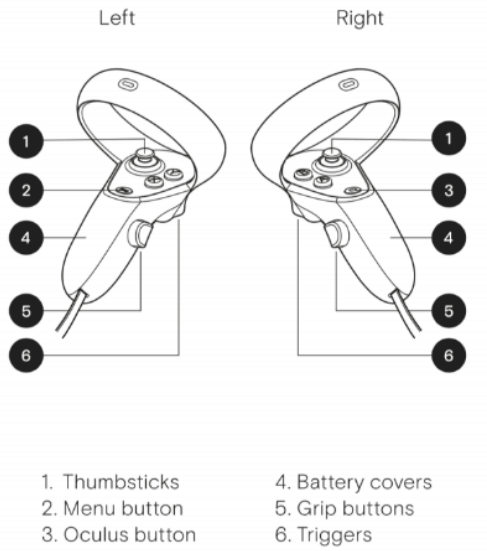
Základní sestava se většinou skládá z headsetu a jednoho ovladače v každé ruce. Existuje několik firem vyrábějících VR systémy, přičemž každá z nich má vlastní typ ovladače. Ovládací prvky ovladačů jednotlivých firem se mohou lišit, avšak díky standardu OpenXR lze některé společné prvky sjednotit do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

- **Trigger** – Je analogová páčka, co detekuje míru stisku, je většinou umístěný pro ovládání ukazováčkem a tvarem i umístěním většinou připomíná spoušť u pistole.
- **Menu** – Je tlačítko, většinou se používá pro spuštění menu ve hře či pro speciální akce.
- **Joystick/Touchpad** – Může mít podobu dotykové plochy s možností mechanického stlačení, nebo klasický joystick s možností stlačení. Většinou je umístěn tak, aby se ovládal palcem a jeho nejčastější použití je pro pohyb ve hře.
- **Grip** – Jsou většinou dvě tlačítka umístěné po obvodu ovladače tak, aby byly zmáčknutelné pevnějším stisknutím ovladače, při jeho klasickém držení.
- **Systém** – Je tlačítko, které otevře systémovou nabídku s nastavením VR. [3]

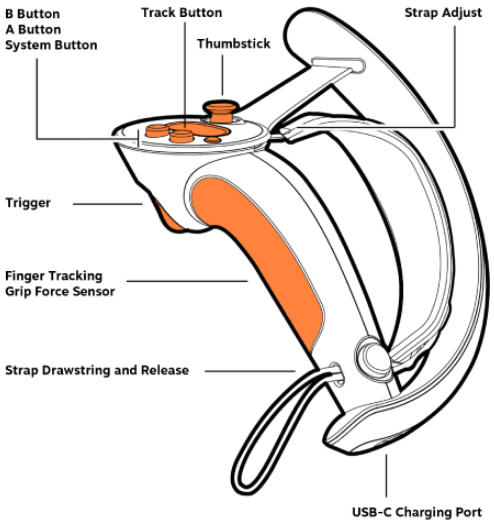
Následující obrázky znázorňují rozložení ovládacích prvků 3 nerozšířenějších značek systémů VR pro PC.



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 – Enchanced Edition

The Wizards – Enchanced 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. [4, 5]

1.4.1 Stručný popis

Ve světě Meliora budete postupovat společně s tajemným hlasem kouzelníka, který vám postupně vysvětluje co se děje a co máte dělat, aby jste zachránili tento svět před jistou zkázou. Postupně prozkoumáváte několik oblastí, každá s několika úrovněmi, ve kterých musíte čelit různým příšerám a řešit logické hádanky. Na vaší cestě budete používat šest kouzel, která postupně objevíte a průběžně můžete vylepšovat. [4]

1.4.2 Mechaniky kouzel

Všechny kouzla se odemknou hned po pár úrovních. Každé z kouzel lze třikrát vylepšit, první dvě vylepšení se vylepšují postupně pomocí Tokenu. Po odemknutí obou vylepšení se zobrazí unikátní výzva pro dané kouzlo, jejím splněním se vylepší potřetí. Pro získání jednoho Tokenu, musíte z úrovní získat 3 Spell Potential body, které získáte za plnění cílů v jednotlivých úrovních. Ve většině úrovní máte za cíl nalezení 3 skrytých magických úlomků a 3 cíle s postupně se zvyšujícím Score, které je spočítáno vždy na konci úrovně. [4]

Všechny kouzla lze rozdělit do několika elementů ohnivé/ledové/bleskové/-fyzické. Kolem některých nepřátel může vzácně být magická aura (štít), který 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. [4]

- **Truhla** – Má na sobě obrázek kouzla, které se na ni má použít pro její odemčení a získání předmětu.
- **Poničená zeď** – Blokuje průchod, jakýkoliv kouzlem ji lze rozbít.
- **Socha** – Střílí na vás, jakýmkoliv kouzlem jí lze rozbít.
- **Vozík** – Jde 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 kouzli zapálit a nechat shořet.
- **Bedny, sudy, ...** – Zničitelné předměty jakýmkoliv kouzlem, občas něco v sobě ukrývají. [4]

V následujícím bodovém seznamu jsou všechny kouzla popsané jak vypadají, jejich použití a jejich jednotlivá vylepšení. Ve většině případů se kouzlo puštěním nebo znovu zmáčknutím Triggeru zruší. [4]

- **Arcane Shield**
  - **Popis** – Magický štít na ruce, dá se jím vyblokovat 3 údery nebo projektily, než se rozbije.
  - **Použití** – V levé ruce zmáčknutím Triggeru a pohybem zleva doprava.
  - **1. Vylepšení: Deflection** – Nepřátelské střely se budou odrážet zpět.
  - **2. Vylepšení: Arcane Bullwark** – Zvýšení výdrže štítu, než se rozbije.
  - **Výzva** – Zabití nepřítele do 1 sekundy, poté vyblokování jeho útoku.
  - **3. Vylepšení: Aegis Throw** – Štít se může hodit a způsobit neutrální poškození, puštěním Triggeru.

- **Fireball**
  - **Popis** – Ohnivá koule, která se hází.
  - **Použití** – V pravé ruce dlaní dolů zmáčknutím Triggeru a následným otočením ruky dlaní vzhůru v malém oblouku.
  - **1. Vylepšení: Ignite** – Fireball zapaluje nepřátele a způsobuje jím tím průběžné poškození.
  - **2. Vylepšení: Empowered Fireball** – Při ponechání Fireballu v ruce delší dobu se zvýší jeho poškození a rychlost kterým letí.
  - **Výzva** – Zasáhnutím nepřítele dvakrát za jednu sekundu pomocí plně nabytého Fireballu.
  - **3. Vylepšení: Hellfire** – Zasáhnutí zapáleného nepřítele zapálí ostatní nepřátele v blízkosti.
- **Pyroblast**
  - **Popis** – Výbušná ohnivá koule, která se hází a vybuchne při dopadu.
  - **Použití** – S oběma rukama před sebou, souběžným stisknutím obou Triggerů a pohnutím dolů.
  - **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** – S oběma rukama před sebou, souběžným stisknutím obou Triggerů a překřížením rukou. Šípy se získají s prázdnou rukou na zádech a stisknutím Triggeru, poté natáhnutím luku a vystřelením puštěním Triggeru.
  - **1. Vylepšení: Frost Shard** – Nepřátele 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 nepřátelům na střední vzdálenost.
  - **Použití** – S oběma rukama u sebe, souběžným stisknutím obou Triggerů, posunutím rychle dopředu a puštěním 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 jím poškození.
  - **Použití** – S oběma rukama před sebou, souběžným stisknutím obou Triggerů, levou rukou nakreslením C zespodu nahoru a pravou rukou přesně obráceně nakreslením vodorovně převrácené C zhora dolů.
  - **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: Arcabe 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. [4]

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 vám pocit, že vážně ovládáte magii.
  - Nápadité využití magie k interakcí 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 zvyšuje komplexnost postupně, čímž snižuje náročnost osvojení jednotlivých částí kouzel a posiluje pozitivní pocit z pokroku.
  - Tokeny dávají důvod snažit se splnit cíle úrovní a projít úroveň s co nejlepším hodnocením, což přispívá k zájmu se perfektně naučit všechna kouzla.

- **Nevýhody**
  - Malá rozmanitost nepřátel, už v polovině hry si zvyknete na všechny typy a přestanete je brát jako výzvu.
  - Magií interaktivní předměty, které se nemají jen rozbít jsou ve hře jen vzácně, což je škoda, jelikož pouhé rozbíjení věcí se rychle omrzí.
- **Poznatky**
  - Vylepšování kouzel, pro které musíte splňovat výzvy pomocí magie a tím motivovat hráče pro prozkoumání všech vlastností jednotlivých kouzel.
  - Ovlivňování nepřátel i status efekty, oproti pouhému poškození, významně zvyšuje hratelnost.
  - Využití zajímavějších interakcí s předměty, než je pouze jejich rozbíjení, snižuje repetetivitu 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. [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. Znovu s tajemným hlasem kouzelníka budete bojovat proti zapovězeným silám, zběhlým kouzelníkům a dalším stvořením, která se vám postaví do cesty při vaší snaze zabránit konci světa, tak jak ho znáte. Postupně projdete několik oblastí a ovládnete 11 kouzel, které vám pomůžou v boji a dalším nástrahám na vaší cestě. [6]

1.5.2 Mechaniky kouzel

Kouzla se odemknou postupně postupem v příběhu. Žádné vylepšení kouzel ve hře není. Výjimkou jsou kouzla, které lze seslat pomocí vytvořeného jiného kouzla v ruce, ale jelikož ve hře jsou popsány zvlášť a fungují jinak zcela samostatně, tak je počítám samostatně. [6]

Každé z kouzel způsobuje elementální (led/oheň/arcane/blesk/radiant) poškození. Existuje několik druhů nepřátel s elementálním typem, pokud je na ně zaútočeno stejným elementem je poškození snížené a níže popsané stavy nenastanou. Různé typy poškození mohou ve správné kombinaci ovlivnit nepřátele a udělovat jím dodatečné poškození. Jednotlivé kombinace je možné popsat pomocí několika stavů podle tabulky 1.1. Normal je defaultní stav, ve

kterém se všichni nepřátelé nacházejí. Po několika sekundách v jiném stavu se nepřítel vrátí zpět do stavu Normal. [6] Většina stavů jsou přechodné, pouze

Původní stav	Typ poškození	Výsledný stav
Normal	ledové	Chill
Chill	ledové	Frozen
Frozen	neohnivé	Shatered
Normal	ohnivé	Warm
Warm	ohnivé	Burn
Chill/Fozen	ohnivé	Wet
Wet	ohnivé	Normal
Normal	bleskové	Shocked

Tabulka 1.1 Tabulka přechodů stavů

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 níže. [6]

- **Burn** – Nepřítel dostává poškození v krátkých intervalech.
- **Frozen** – Nepřítel zamrzne do ledové sochy a nemůže se pohnout ani zaútočit.
- **Shatered** – Ledová socha je roztříštěna a nepřítel instantně zemře.
- **Shocked** – Nepřítel se třese a je znehybněn. [6]

Kromě boje s nepřáteli, kouzla mají využití na překonání různých překážek, hádanek, či rozbíjení různých objektů. [6]

- **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 zchladit a následně roztříštit pomocí ledového a arcane kouzla respektivně.
- **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, ...** – Doplňkové předměty, které se dají zničit všemi kouzly. [6]

V následujícím bodovém seznamu jsou všechny kouzla popsané jak vypadají, jejich využití, gesto vytvoření a gesta použití. Ve většině případů se kouzlo puštěním tlačítka (Trigger/Grip), kterým se vytvořilo použije nebo zruší, pokud by hráč chtěl zrušit vytvořené kouzlo a chce použít jiné. [6]

- **Fireball**
  - **Popis** – Ohnivá koule, která se hází.
  - **Využití** – Hozením způsobuje ohnivé poškození.
  - **Gesto vytvoření** – Jednou rukou dlaní dolů zmáčknutím Triggeru a následným otočením ruky dlaní vzhůru v malém oblouku.
  - **Gesto použití** – Mávnutím ruky, jako při hodu a puštěním Triggeru. Je možnost Fireball po vypuštění zase chytnout stisknutím 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, stisknutím obou Gripů.
  - **Gesto použití** – Oddálením rukou od sebe se začne tvořit trhlina směrem, tam kam ukazují ruce. Puštěním 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, zmáčknutím Gripu a pohnutím horizontálně ruky k sobě.
  - **Gesto použití** – Pohybem ruky od sebe a puštěním Gripu. Rychlejší pohyb udělá kouzlo silnější.

- **Arcane Shield**
  - **Popis** – Magický štít na ruce, kterým se dá 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 stisknout Trigger a pohnout jí ze strany doprostřed.
  - **Gesto použití** – Štít pohnout do cesty projektilům nebo úderům pro jejich vyblokování. Štít se zahodí mávnutím ruky, jako při hodu a puštěním 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, stisknutím 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 udělují ledové poškození. Po použití všech 5 šípů se luk rozbije.
  - **Gesto vytvoření** – S rukama před sebou, stisknutím obou Triggerů a posunutím rukou přes sebe do kříže. Lze pustit Trigger na ruce bez luku (dominantní ruka).
  - **Gesto použití** – Stisknutím Triggeru dominantní ruky poblíž tětivy se vytvoří šíp. Puštěním Triggeru po natáhnutí tětivy a zamířením stejně jako v realitě, vystřelíte šíp.

- **Ice Sabers**
  - **Popis** – Šavle (v každé ruce jedna) z ledu, které se sestaví z materiálu, ze kterého byl Frost Bow.
  - **Využití** – Hozením nebo úderem způsobí poškození a instantní zmražení nepřátel do stavu Frozen. Jakýmkoliv použitím se hned roztříští.
  - **Gesto vytvoření** – S vytvořeným kouzlem Frost Bow a rukama u sebe, stisknutím obou Gripů s následným odtažením rukou od sebe.
  - **Gesto použití** – Máchnutím na nepřítele nebo zahozením mávnutím ruky, jako při hodu a puštěním 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 stisknutím Triggeru.
  - **Gesto použití** – Puštěním 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 Lighting a rukama u sebe, stisknutím obou Gripů.
  - **Gesto použití** – Odtažením 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), zmáčknutím Triggeru a posunem ruky před sebe.
  - **Gesto použití** – Rychlým pohybem ruky směrem, tam kam chcete a puštěním 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, stisknutím obou Gripů, počkáním na nabití a poté oddělením rukou od sebe. V závislosti na délce nabití se vytvoří 2 až 10 krystalů.
  - **Gesto použití** – Dotykem jednotlivých krystalů, je vystřelíte. [6]

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 vám pocit, že vážně ovládáte magii.
  - Kombinace kouzel, jako například zmražení a následné roztříštění, 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**
  - Nevyužitím všech kombinací mezi kouzly, ztrácí systém celistvost, nevyužívá plného potenciálu této mechaniky a může být i matoucí.
  - Truhly s Runy, jsou skvělý nápad, ale chtělo by to přidat další varianty hádanek pro hráče s využitím kouzel, aby se to tak rychle neomrzelo.
  - Přez jednoduchost a přehlednost gest se nepodaří v zápalu souboje vytvořit chtěné kouzlo ve velkém množství případů. Konkretně se často stalo místo Arcane Shieldu vytvořil Fireball, což v souboji proti několika nepřátelům znamená smrt.

- **Poznatky**
  - Kombinace kouzel zvyšuje jejich rozmanitost a využití, ale je potřeba vytvořit všechny možné kombinace.
  - Použití vytvořených kouzel pro vytvoření jiného souvisejícího, zjednoduší učení a zminimalizuje obtížnost zapamatování kouzel.
  - Důležitost důkladného popisu a přesnosti detekce gest, pro zamezení seslání špatného kouzla.
  - Vytvořit více variant hádanek řešených kouzly, jelikož 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. [7, 5]

1.6.1 Stručný popis

V této hře hráč nemá žádný konkrétní cíl, ale má volnost zkoumat okolí a experimentovat s různými předměty s cílem řešit výzvy a záhady, které tento svět nabízí. Hra začíná v nejvyšší místnosti věže, která kdysi patřila mocnému kouzelníkovi a nyní je plná tajemných předmětů po něm zanechaných. Během téměř každé akce ve věži hráči radí tajemná lebka s jménem Skully, ležící na stole. Později je možné s touto lebkou mluvit a dávat jí povely na ovlivňování prostředí. Hráč může postoupit dále tím, že splní tři úkoly zobrazené na pečeti u vstupu a poté se pečeť zlomí a hráči se odemkne vstup na dvůr. Stejným postupem se hráči postupně dostane 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. [7]

1.6.2 Mechaniky kouzel

Hra obsahuje 3 rozdílné mechaniky, které se od sebe liší. 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 kombinací. Nelze kombinovat více než jednu ingredienci z jedné skupiny, přičemž přidáním všech 3 ingredienci 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. [7]

- **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. [7]

Zbývajících 5 kouzel popsaných níže, má své specifické použití. [7]

- **Magnetize** – Zmáčknutím Triggeru přitáhne věci k ruce jako magnet a po puštění je od sebe odstřelí.
- **Conductor** – Pro vše v okolí přestane platit gravitace, zmáčknutím Triggeru a pohybem ruky se totožně pohnou všechny okolní předměty. Puštěním budou předměty pokračovat v pohybu, ve kterém se pohybovali jako bez gravitace.
- **Puppeteer** – Zmáčknutím Triggeru a chycením předmětu se zaznamená jeho pohyb, až do puštění a začne se pohybovat stejně jako s ním bylo pohnuto od začátku na konec na začátek a stále dokola.
- **Air Drawing** – Pomocí stisknutí Gripu se začne kreslit čára do vzduchu až do puštění Gripu a kresba se stane objektem. Triggrem lze uchopit a pustit jako ostatní předměty.
- **Fireball** – Zmáčknutím Triggeru se vytvoří ohnivá koule v ruce, poté máchnutím a puštěním Triggeru se dá vyhodit, zničí či poškodí to na co dopadne. [7]

Druhá mechanika se nazývá Nature Magic. Mimo vnitek věže, kde je blokována, funguje vždy a všechny její kouzla jsou hned ze začátku odemknutá. Hráč, pokud hraje hru poprvé, neví jak ji ovládat či kolik kouzel obsahuje a až postupem ve Fortress postupně sbírá plakáty s návodem, jak jednotlivá kouzla používat. [7]

- **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** – Stiskem Triggeru v určené ruce a posunutím rukou přes sebe do kříže s určenou rukou jako vnější. Puštěním 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 částečky zmizí po kontaktu, ale po chvíli se obnoví.
  - **Gesto** – Stiskem Triggeru a rychlým posunutím ruky dolů. Puštěním se zruší.
- **Nature Magic**
  - **Vzhled** – Malé fialové částice poletující kolem ruky.
  - **Použití** – Používá se k ostatním kouzlům z přírodní magie.
  - **Gesto** – Stisknutím Triggeru a posunutím ruky k tělu s dlaní směrem k sobě. [7]

Pro všechny následující kouzla je potřeba nejdříve vytvořit Nature Magic v ruce (v popisu Gesta je myšlena tato ruka). [7]

- **Spectral Orb**
  - **Vzhled** – Koule z bílých částic, která se chová jako bowlingová koule.
  - **Použití** – Koule poškodí vše do čeho narazí, než za chvilku zmizí.
  - **Gesto** – Zmáčknout Trigger s rukou dole a máchnout jí jako při hodu s bowlingovou koulí s puštěním 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** – Zmáčknout Trigger, máchnout rukou směrem a rychlostí kam chcete aby částice letěly s puštěním 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 vše v okolí.
  - **Gesto** – Dát ruce k sobě,fialové částice začnou zrychlovat svůj pohyb až zčervenají, zahodí se stisknutím a puštěním 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 stisknutím a puštěním 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** – Stisknout Trigger, mávat rukou v horizontálních kruzích, až se začne dělat malé tornádo, zahodí se puštěním 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** – Stisknout Trigger, rychle pohybovat rukami u sebe střídavě nahoru a dolů, než se vytvoří bleskové výboje, zahodí se puštěním Triggeru. [7]

Třetí a poslední mechanikou je ”Aladinův prsten”, který dovoluje komunikovat anglickým jazykem a tím dávat povely pomocí rozpoznávání hlasu. Pouze některé sousloví jsou implementované a v případě, že neznáte konkrétní slova nebo se nerozpoznaly správně, tak se nic nestane. U některých povelů existuje několik alternativ, které udělají to samé. Jedním z povelů je například přepínání, mezi 12 lektvarovými kouzly slovy: ”Give me …”, kde doplníte název kouzla a deaktivujete ho pomocí slov: ”Clear magic”. Zbytek povelů umožňuje ovládat realitu (měnit gravitaci, velikosti objektů nebo je vytvářet, …). V následujícím výčtu jsou popsány základní povely, nutno brát v potaz, že většinu částí jde obměnit či pozměnit například ”Make me tiny”změnit na ”Make me giant”nebo ”Make that bouncy”lze nahradit za ”Make this bouncy”. [7]

- **”Make me tiny”** – Zmenší hráče.
- **”Make me healthy”** – Uzdraví hráče od všeho obdrženého poškození.
- **”Make me increadibly fast”** – Zvýší rychlost chození.
- **”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 předmět na který se hráč dívá nebo ukazuje, odrážel od všeho.
- **”Make this sticky”** – Udělá předmět na který se hráč dívá nebo ukazuje lepivým.
- **”Make everything normal”** – Všechno se vrátí (jen jejich vlastnosti) do původního stavu.
- **”Create a granade”** – 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. [7]

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 a sesílat kouzla, umožňuje hráči se více propojit se 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, kde 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.
- **Poznatky**
  - Odemykání kouzel pomocí míchání přísad či jiných surovin je skvělý 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. [8, 5]

1.7.1 Stručný popis

Hra se odehrává v režimu 1v1 nebo 2v2 s ostatními hráči online, každý tým se 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 svůj unikátní účinek, a hráči si musí strategicky vybírat, jaká kouzla použít. [8]

1.7.2 Mechaniky kouzel

Před každým zápasem si hráč vybere 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. [8]

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 sesláno. Potvrdit sesílání lze stiskem a podržení 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 zvlášť 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 dobíjí magickou střelou po každém zásahu nepřítele nebo nepřátelských jednotek. [8]

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 neliší v způsobu sesílání, v následující části jsou popsané jen 3 kouzla pro ukázku mechanik. [8]

- **Swordstorm Spell**
  - Cena pro seslání je 400 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**
  - Cena pro seslání je 200 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 Command

- **Divine Shield Spell**
  - Cena pro seslání je 200 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. [8]

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 obrazců 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 číslel.
- **Nevýhody**
  - Způsob sesílání kouzel pomocí kreslení obrazců není vhodný pro akční žánry her, zejména ty, ve kterých je nutné se rychle pohybovat a bojovat proti více protivníkům najednou.
- **Poznatky**
  - 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 obrazců 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 výsledky analýz jednotlivých her.

První hra The Wizards Enchanced Edition obsahuje 6 kouzel, které lze postupně vylepšovat. Mezi přednosti patří jednoduchý bojovým systém, společně s různorodým použitím kouzel pro souboj a interakcemi s předměty. Dále má důmyslný systém vylepšování kouzel s motivací hráče k dosažení maximálního hodnocení v úrovních. Hlavní nedostatky této hry je nedostatečná rozmanitost nepřátel a malý počet výskytu zajímavě interaktivních předmětů.

V pokračování The Wizards Dark Times Brotherhood je celkem 11 kouzel, každé patřící do jednoho z několika elementů. Mezi hlavní výhody patří mecha-

nika vylepšení základních kouzel na kouzla jiná, což umožňuje hráčům snadné zapamatování 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, respektivně. 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 sesílají kreslením odpovídajících obrazců do prostoru a následně ohodnocených podle jejich velikosti a přesnosti. Skvělým nápadem je podle tohoto hodnocení dynamicky vylepšovat různé vlastnosti seslaných kouzel a strategické omezení počtu použitelných typů kouzel pro jeden zápas. Kreslení obrazců pro sesílání kouzel dává téměř nekonečné možnosti přidávat další, avšak tento systém není vhodný pro akční hry, kde je tento styl sesílání moc pomalý a nedovoluje volný pohyb při soubojích.

1.9 Výsledek analýzy

Tématem této práce je vytvořit systém pro sesílání kouzel pomocí gest, přesto se v této sekci řeší i pár témat, které se týkají tohoto tématu pouze okrajově a spíše se hodí pro budoucí zapracování tohoto systému do her. Tyto vedlejší témata jsou důležité pro odůvodnění některých rozhodnutí, ale nebudou zapracovány do výstupu práce. Každé řešení her analyzovaných výše má své výhody a nevýhody, které jsou v této části podrobněji rozebrány a ohodnoceny, podle jejich přínosnosti k návrhu autora.

- V těchto hrách se používaly celkem 3 různé způsoby sesílání kouzel: jednoduchá gesta, kreslení složitých obrazců a rozpoznávání hlasu.
  - Technologie rozpoznávaní hlasu se do návrhu nehodí, přestože to je zajímavý doplňující způsob ovládání magie, tak není dostatečně spolehlivý, je komplikovaný na provedení a někteří hráči mohou mít problémy se správnou výslovností anglického jazyka.
  - Kreslení složitých obrazců do návrhu nesedí, i když umožňuje velký počet kouzel, tak je vysoce nepraktické pro akčnější hry, kde se hráč potřebuje pohybovat rychle v prostoru a vyhýbat útokům v reálném čase.

  - Jednoduchá gesta jsou nejlepším řešením, jelikož dovolují rychlou reakci na různé situace, jsou jednoduché na zapamatování a umožňují souběžné sesílání kouzel oběma rukama zároveň.

- Mechanika vytváření kouzel skládáním několika gest po sobě (vytvořeno gestem z jiného kouzla) se hodí do návrhu, protože přidává hloubku společně s vyšším počtem kouzel a zároveň zachovává jednoduchost gest.
- Za zvážení stojí i kombinování dvou kouzel do jednoho, jelikož.
- Škálování síly kouzel dynamicky podle přesnosti vykonání gesta přidává další způsob do návrhu, který umožní trénováním vylepšit sílu kouzel, čímž zároveň navádí hráče, aby se snažil co nejvíce naučit jednotlivé gesta a je imerzivnější, než jen zvýšení čísla síly kouzla.
- Elementální magie je další prvek, který dovoluje hru nadále rozšířit a společně s ním se otevírá možnost různých statusových efektů, které dále rozšiřují způsoby souboje a dávají další možnosti k taktizování.
- Další důležitou metrikou pro hráče je progress a složitost mechanik, které se musí naučit používat proto, aby si hru mohl užít.

  - Omezení kolik které lze najednou používat společně s vyšším počtem kouzel je skvělým nápadem, který zároveň umožňuje vyšší dynamiku a možnosti hraní. Toto zamezuje přehlcení hráče vysokým počtem možností, které by měl k dispozici najednou, ale zároveň umožňuje taktizovat a různí svůj herní styly.
  - Postupné odemykání dalších kouzel také zabraňuje přehlcení hráče informacemi najednou a umožňuje obsah hry dávkovat po menších částech.
  - Mechanika vylepšování kouzel přidává pocit postupu a zároveň umožňuje odměňovat hráče pomocí různých odměn získaných hraním hry. Tímto se odemykají možnosti v budoucnu přidat dalších mechaniky, kterými se může prohloubit uvěřitelnost herního světa a otevřít další možnosti kam hru rozšiřovat. Například se může jednat o crafting systém a získávání surovin (zabíjením monster, těžením, …).

- Interaktivní předměty v prostředí VR jsou obzvlášť důležité, jelikož značně zvyšují uvěřitelnost prostředí a vtahují hráče více do herního světa.

  - Zničitelnost okolních předmětů je v dnešní době téměř nutností.
  - Pro lepší zážitek je potřeba také navrhnout i další typy interakcí, které jsou komplexnější a mají nějaký účel. Například se může jednat o zapalování a uhašení pochodní, či řešení hádanek, které potřebují použití hned několika elementů.

- V poslední řadě je podstatnou částí správné hry dobře zpracovaný a pochopitelný tutoriál. I když hra obsahuje skvělé mechaniky, hráč je nedokáže plně využít a ocenit, pokud o nich neví nebo je neumí správně ovládat.

Kapitola 2
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 aplikace do širokého spektra platforem (PC, mobilních zařízení, herních konzole, …) 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]

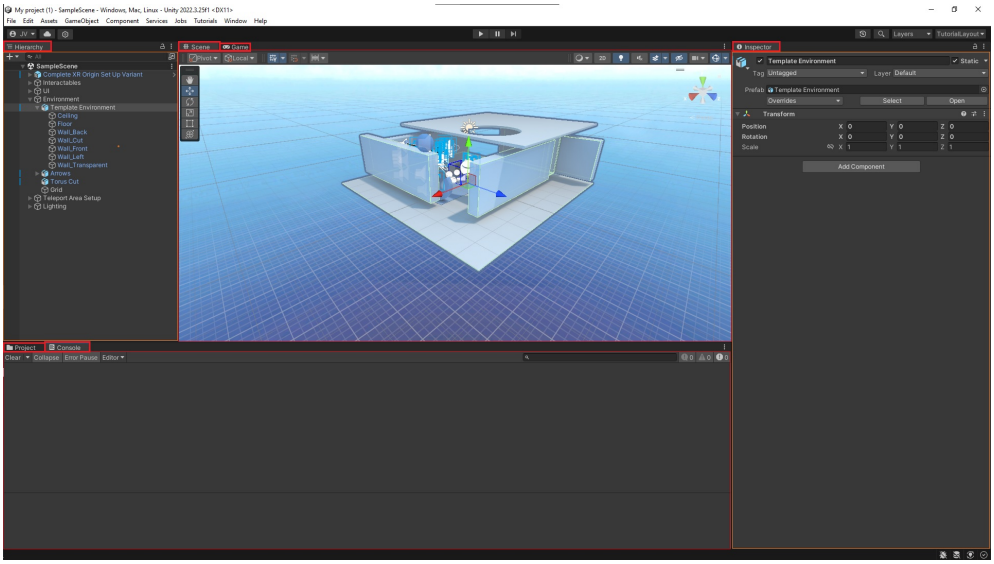
Společně s nejznámější platformou pro tento obsah nazývající se Unity Asset Store [10], nabízí širokou škálu předem vytvořených modelů, textur, zvukových efektů, skriptů a dalších věcí, které vývojáři mohou využít ke zrychlení svého vývojového procesu. Kromě toho díky rozsáhlé komunitě vývojářů existuje 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 také komponentem a společně se širokou škálou dalších komponentů 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 vašich 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. 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 ve stručnosti popsány v seznamu níže a zvýrazněny na následujícím obrázku 2.1.

- **Hierarchy** – Zobrazuje všechny objekty v aktuální scéně jako text v hierarchickém seznamu s možností skrytí podobjektů.
- **Inspector** – Zobrazí komponenty nakliknutého objektu, které se zde mohou upravit, přidat či odstranit.
- **Scene** – Vizualizuje rozmístě objekty v aktivní scéně a umožňuje s nimy 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]



Obrázek 2.1 Základní rozmístění oken v Unity (snímek obrazovky) [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 zde zkráceně popíšu. [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 UV mapu, texturami a materiály. V nástroji se pracuje UV mapou, která 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 umožnit modelům se pohybovat a vytvářet z 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]

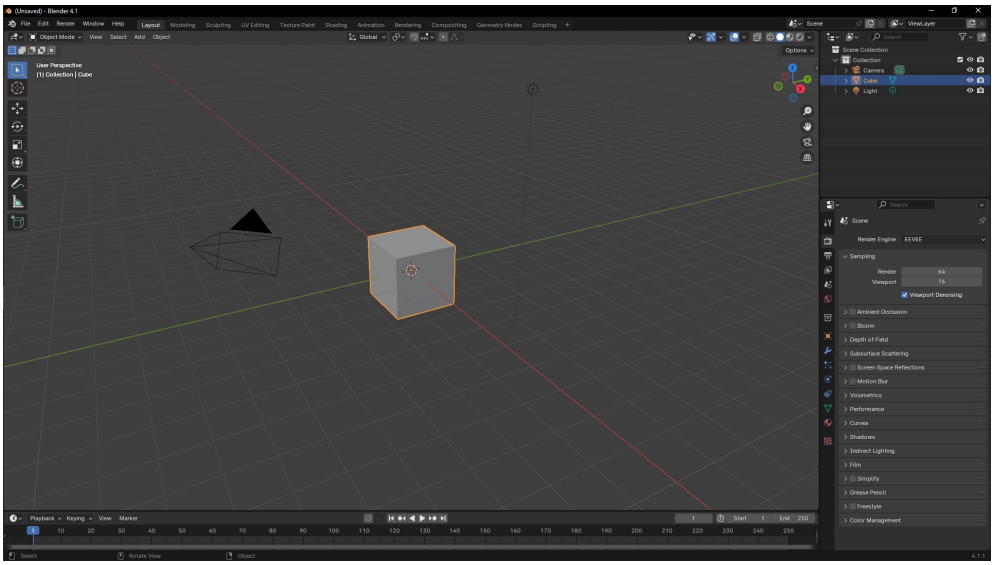
Uživatelské rozhraní Blenderu je tvořeno pomocí několika základních pracovních prostřední (Workspace), které se dají přepínat pomocí tabů na horní liště. Každý Workspace je rozdělen na několik optimálně rozdělených a nastavených editorů, které dohromady tvoří základní sestavu nástrojů pro danou činnost. 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 ukázku 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. 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 ukázku 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]

Kapitola 3
Návrh

Základní funkční požadavky nutné pro funkční systém sesílání kouzel pomocí gest:

- Vytvoření několika krystalů s několika elementy (oheň, led, …).
- Několik kouzel s různým použitím, každé z nich lze vytvořit v každém elementu.
- Jednoduchá gesta na vybírání krystalu.
- Jednoduchá gesta pro sesílání kouzel.
- Základní předměty interagující s kouzly.

Dodatečné návrhy pro budoucí rozšíření a zapracování tohoto systému hry (nejsou součástí realizace této práce), plynoucí z analýzy a názoru autora:

- Vytvoření více typů elementálních krystalů (společně s omezeným počtem míst pro krystaly v každé ruce).
- Vylepšení krystalů pro silnější magii a dodatečné vlastnosti.
- Systém pro crafting a získávání surovin.
- Kombinování dvou různých elementů v pravé a levé ruce do jednoho kompozitního elementu.
- Speciální vlastnosti elemetů na nepřítele (led – zpomalí, oheň – zapálí a uděluje poškození, …).
- Hierarchie základních a kompozitních elementů (voda je slabá proti ledu, …).
- Typy nepřátel se slabinami či odolností vůči specifickým typům elemetů a útoků.
- Předměty se složitější interakcí s kouzly (magické zámky prolomitelné pomocí určité magie, hádanky s nimi spojené, …).

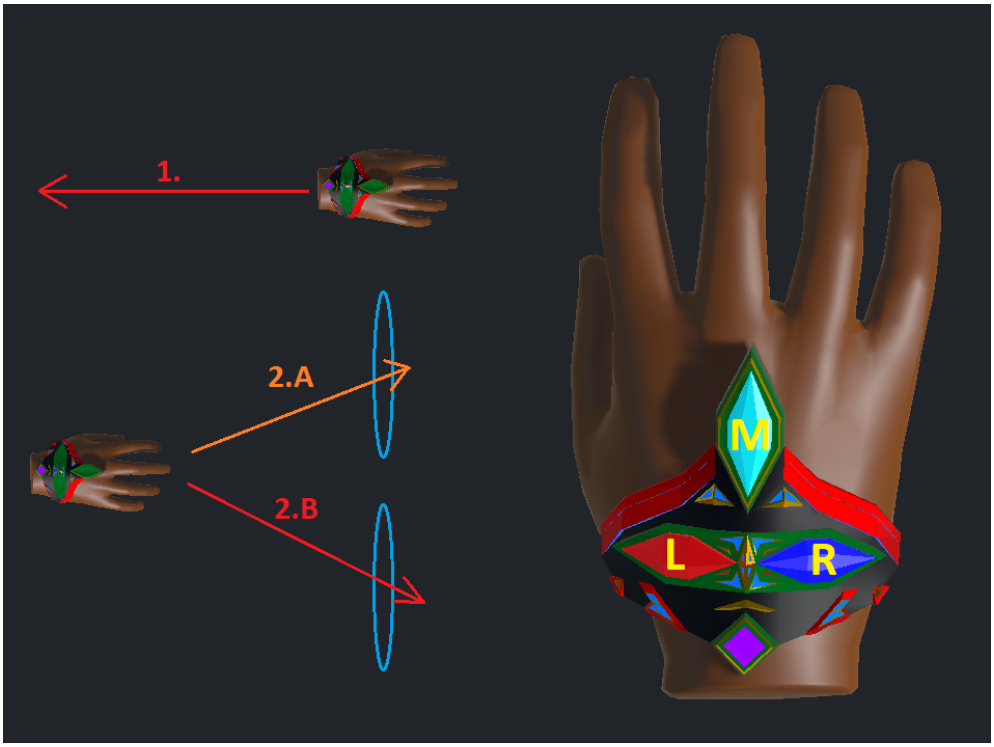
- Více typů kouzel společně s rozšířením jejich použití (štít se bude dát zahodit a odstrčí tím nepřátele).
- Kompozitní kouzla tvořené z již vytvořených kouzel nebo souběžně oběma rukama s zároveň vybraným stejným živlem.
- Ovlivnění síly kouzla, podle přesnosti provedení gesta.
- Dovednostní stromy pro odemykání a vylepšování jednotlivých kouzel.
- Tutoriál zapracovaný přímo do hry.

Pro přehlednost celého procesu sesílání kouzel pomocí gest se celý proces rozdělí na 3 základní části. Toto rozdělení nejen zjednoduší plánování a realizaci, ale zároveň rozkouskováním této mechaniky na menší části usnadní celý proces tak, aby ho hráč snadněji pochopil a ovládl.

1. **Zvolení elementu** – Zvolení krystalu, který určuje element, ze kterého se budou vytvářet kouzla.
2. **Vytvoření kouzla** – Vykonání gesta pro tvorbu jednotlivých kouzel.
3. **Použití kouzla** – Ovládání vytvořených kouzel.

3.1 Zvolení elementu

Pro zvolený rozsah této bakalářské práce stačí 3 základní elementy (led, oheň, voda) s tím, že se má počítat s případným přidáním dalších v budoucnu. Každá ruka má 3 pozice pro elementální krystaly. Tyto pozice se rozdělují na 1 aktivní (M), který se aktuálně používá pro Vytváření kouzel a 2 vedlejší (L + R), které slouží jako rezervní dle žlutého písma v pravé části obrázku 3.1. Zároveň na něm jdou vidět všechny 3 typy krystalů (M – led / L – oheň / R – voda). Krystaly z vedlejších pozic se dají prohodit s krystalem v aktivní pozici pomocí gesta zakresleného na levé části obrázku 3.1. Toto gesto se skládá z 1. kroku, ve kterém je ruka ve vodorovné pozici dlaně, stiskne se Grip a posune se podélně směrem k uživateli. Celé gesto se přeruší při jakémkoliv natočení ruky v 1. kroku nebo při puštění Gripu v jakékoliv fázi a je nutné opakovat od začátku. V 2. kroku před rukou zobrazí dva magické kruhy s barvou jednotlivých krystalů ve vedlejších pozicích, následným posunem ruky doprostřed jakéhokoliv z nich se krystal na stejné straně prohodí s aktivním (2.A -> L / 2.B -> R).

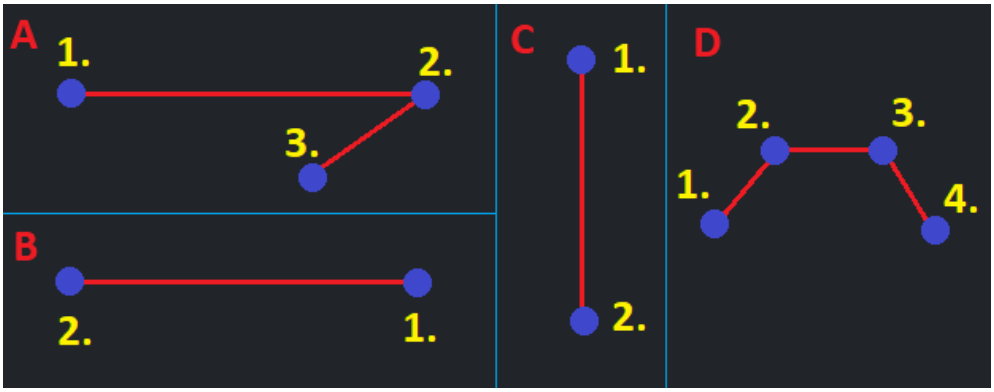


Obrázek 3.1 Rozmístění krystalů a gesto pro jejich prohození

3.2 Vytvoření kouzel

Pro zvolený rozsah bakalářské práce stačí 4 základní kouzla (štít, meč, kulatý projektil, ostrý projektil) s tím, že se má počítat s případným přidáním dalších v budoucnu. Každé z kouzel začne tím, že hráč zmáčkne Trigger, začne pohybovat rukou do tvaru jednoho z kouzel a poté pustí Trigger. Při držení Triggeru se kreslí čára do vzduchu, která zobrazuje dosavadní pohyb ruky od stisknutí. V případě že se nakreslená čára nepodobá žádnému kouzlu, nic se nestane, v opačném případě se vytvoří dané kouzlo s elementem podle krystalu, který je v tu dobu na aktivní pozici. Pořadí určující směr kreslení je očíslováno žlutými číslicemi a jednotlivé typy kouzel jsou označeny červeným písmenem.

- **A** – ostrý projektil
- **B** – štít
- **C** – meč
- **D** – kulatý projektil



Obrázek 3.2 Základní gesta kouzel

3.3 Použití kouzel

Použití každého kouzla se liší, proto je každé z navrhnutých kouzel popsáno zvlášť v seznamu níže.

- **Meč** – Vytvoří se meč sevřený v ruce. Stejně jako v realitě uděluje poškození pomocí fyzického kontaktu. Stiskem Triggeru a jeho puštěním se toto kouzlo zruší.
- **Štít** – Vytvoří se poloprůhledný štít ve tvaru zaobleného obdélníku na výšku. Jeho pozice závisí na úhlu mezi rukou a headsetem. Blokuje nepřátelské útoky, které do něho narazí. Stiskem Triggeru a jeho puštěním se toto kouzlo zruší.
- **Ostrý projektil** – Vytvoří se ostrý projektil nad rukou. Namířením, stiskem Triggeru a jeho puštěním se vystřelí směrem dle své orientace, po zásahu uděluje poškození.
- **Kulatý projektil** – Vytvoří se kulatý projektil pod rukou. Stiskem Triggeru, pohybem ruky jako při hodu a puštěním Triggeru se vyhodí směrem hodu, po zásahu uděluje poškození.

3.4 Interaktivní předměty

Pro zvolený rozsah bakalářské práce budou stačit 2 předměty, které interagují s kouzly.

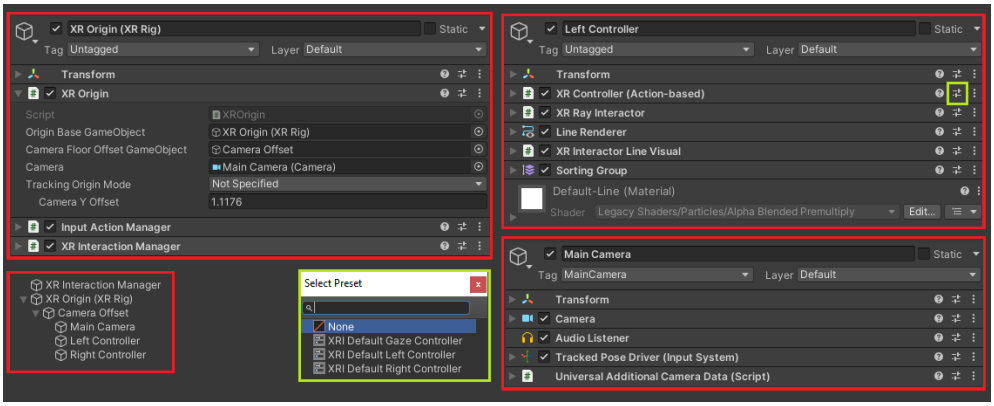
- **Pochodeň** – Dá se zapálit útokem pomocí ohně a uhasit útokem pomocí ledu nebo vody.
- **Zničitelná krychle** – Vodním útokem se zvětší, ledový útok ji odstrčí a ohnivý se zničí.

Kapitola 4
Implementace

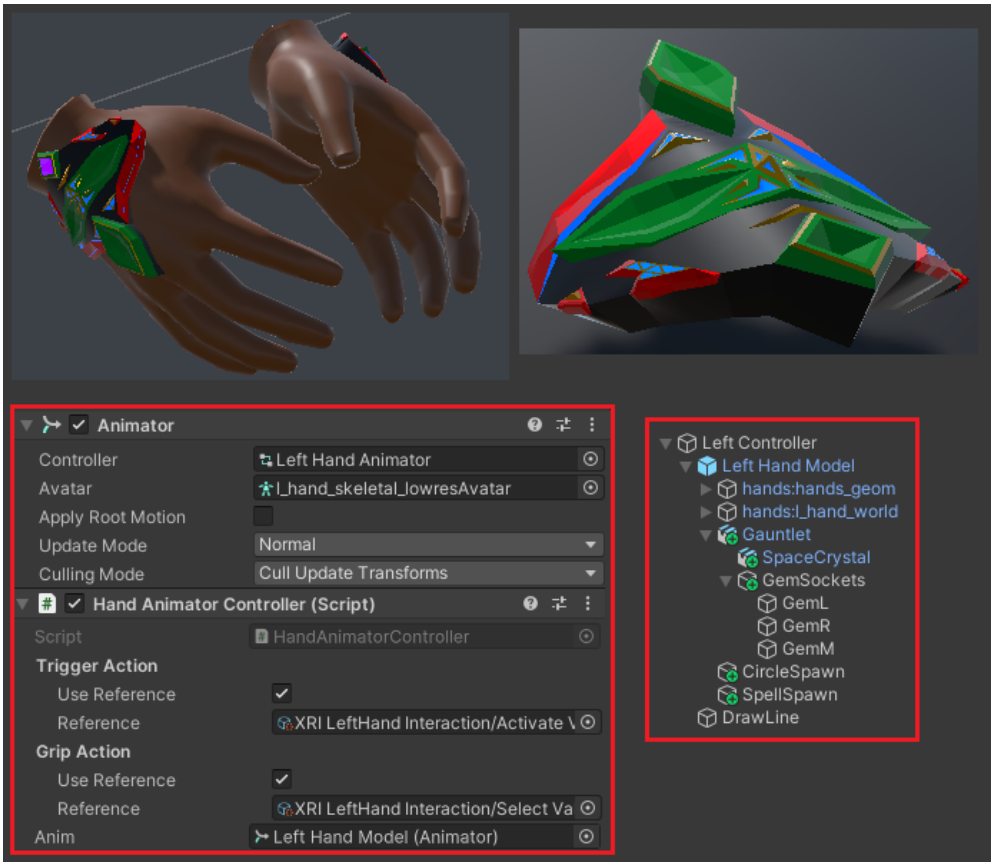
4.1 Vytvoření základního prostředí ve VR

V Unity (2022.3.25f1) vytvořím projekt se šablonou nazvanou *VR (Core)*. Tato šablona obsahuje už předem nainstalované rozšíření OpenXR, které sjednocuje všechny systémy, čímž umožňuje s nimi jednoduše pracovat bez potřeby definovat interakce s aplikací pro každý systém zvlášť. Pro testování jsem využil automatického napojení na SteamVR z aplikace Steam, která mi můj projekt automaticky napojila na můj systém pro VR HTC Vive. Dále šablona obsahuje základní scénu a prefaby s různými ukázkami použití OpenXR, z těchto assetů použiji jen některé části pro UI. Vytvořím novou scénu, v ní odstraním kameru, vytvořím nový objekt z menu *3D Object* > *Plane*, který použiji jako podlahu pro pohyb, a *XR* > *XR Origin (VR)*, který vytvoří základ pro fungování VR ve scéně. Přesunu komponent *XR Interaction Manager* ze stejnojmenného objektu na objekt *XR Origin VR*, pro zpřehlednění. V komponentě *XR Origin*, je třeba nastavit aby se výška hráče počítala od podlahy v proměnné *Tracking Origin Mode*. Dále v obou ovladačích nastavím komponentu *XR Controller (Action–based)* na jejich presety pro levý ovladač a pravý ovladač, *XRI Default Left Controller* a *XRI Default Left Controller* respektivně, které nastaví snímání vstupů z ovladačů (obrázek 4.1).

Následně jsem v komponentě *XR Controller (Action–based)* vymazal defaultní model z proměnné *Model Prefab*, místo kterého jsem vložil do *Model model* vložil svůj model, který jsem předtím zařadil jako podobjekt do každé ruky. Tento model se skládá z animovaných rukou pro Oculus [12], mého vlastního modelu *Gauntlet* a několika prázdných podobjektů, které jsou napozicované pro další účely rozebrané níže. Vytvořil jsem scrip *HandAnimatorController* 4.1, který jsem nastavil tak, aby vstupy z ovladače řídily animátor pro animace ruky.



Obrázek 4.1 Základní rozložení objektu XR Origin a jeho komponent



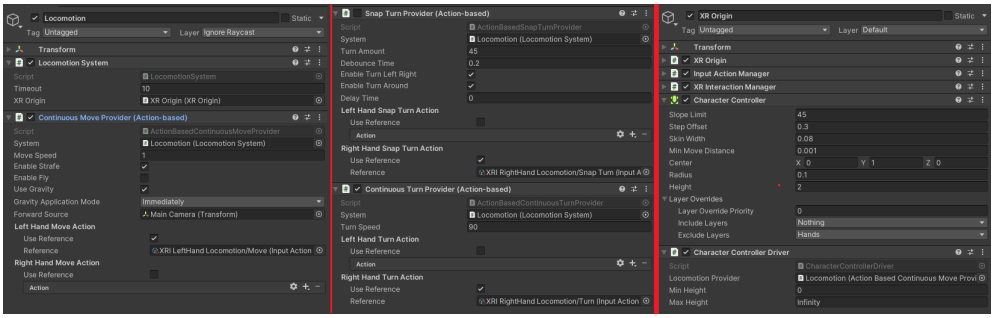
Obrázek 4.2 Modely, script a složení objektu ruky

```
public class HandAnimatorController : MonoBehaviour
{
    [SerializeField] InputActionProperty triggerAction;
    [SerializeField] InputActionProperty gripAction;
    [SerializeField] Animator anim;
    void Update(){
        anim.SetFloat("Trigger", triggerAction.action.
            ReadValue<float>());
        anim.SetFloat("Grip", gripAction.action.ReadValue<
            float>());
    }
}
```

Výpis kódu 4.1 HandAnimatorController.cs

4.1.1 Pohyb a rotace

Pro přidání možnost pohybu a rotace hráče pomocí Touchpadu jsem přidal další prázdný objekt do *XR Origin*, který nazvu *Locomotion* a přidám do něho potřebné komponenty, které pak nastavím tak jak jde vidět na obrázku 4.3. Hlavní komponentou řídící pohybový systém je *Locomotion System* společně s *Character Colider* a *Character Collider Driver*, které bylo potřeba přidat do *XR Origin*. Možnost chůze zařizuje *Continuous Move Provider (Action–based)*, ve kterém jsem nastavil, aby pouze levý ovladač spouštěl chůzi pomocí Touchpadu směrem relativně oproti pohledu Headsetu. Poslední dva komponenty umožňují sekanou a plynulou rotaci (aktivní bude vždy pouze jeden), oba jsou nastavené tak, aby pouze pravý ovladač spouštěl rotaci pomocí pravé a levé strany Touchpadu.

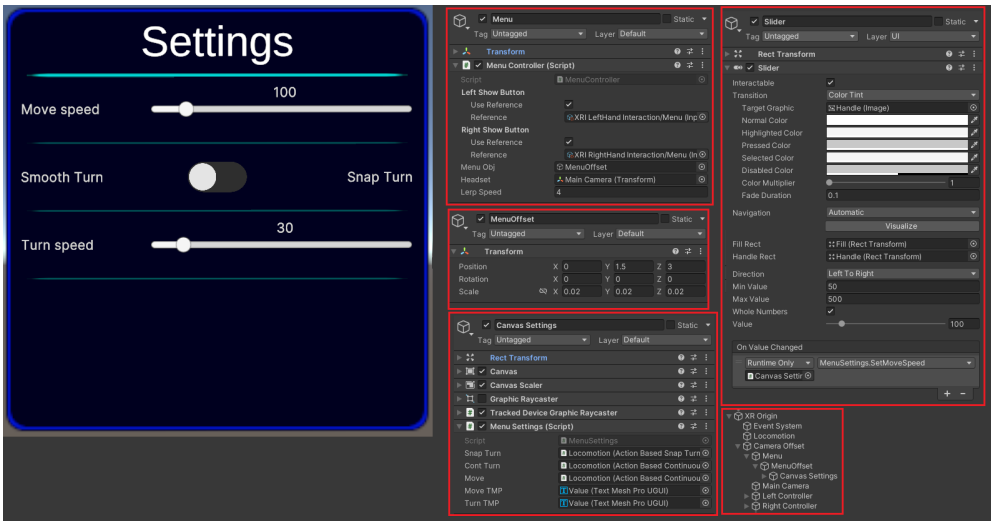


Obrázek 4.3 Nastavení pohybových komponent v objektu Locomotion a XR Origin

4.1.2 Menu

Nejdříve jsem vytvořil prázdný objekt *Menu* do *Camera Offset*, v něm další prázdný objekt *MenuOffset* a toho jsem vytvořil nový objekt z menu *XR > UI*

Canvas, všechny tyto nově vytvořené objekty jsem nastavil do vrstvy (Layer) UI. Společně s ním se ve scéně vytvořil objekt *Event System*, který jsem přesunul do *XR Origin*, zařizující svými komponenty interakce mezi UI a VR. V každé ruce jsem nastavil *XR Ray Interactor*, aby reagoval pouze na UI vrstvu a neukazoval se, pokud nemíří na interaktivní povrch. K objektu *Menu* jsem přidal script 4.3, který obstarává posouvání podle pozice a rotace headsetu, aby bylo menu stále před hráčem a zároveň ovládá jeho skrývání s odkrýváním při stisku tlačítka Menu. Prázdný objekt *MenuOffset* slouží k odsazení vnitřního obsahu do správné polohy. Objekt *Canvas Setting* obsahuje všechny grafické assety jako podobjekty a má k sobě přidaný script 4.2 pro zprovoznění jednotlivých částí v menu. Pro vytvoření designu jsem použil lehce upravené assety ze základní šablony, kde se bude dát nastavit několik věcí ohledně pohybu.



Obrázek 4.4 Vzhled menu a nastavení komponent

```
1  public class MenuSettings : MonoBehaviour
2  {
3      [SerializeField] SnapTurnProviderBase snapTurn;
4      [SerializeField] ContinuousTurnProviderBase contTurn;
5      [SerializeField] ContinuousMoveProviderBase move;
6      [SerializeField] TextMeshProUGUI moveTMP;
7      [SerializeField] TextMeshProUGUI turnTMP;
8      void Start(){
9          UpdateTurnText();
10         UpdateMoveText();
11     }
12     public void SwitchTurn(bool snap){
13         snapTurn.enabled = snap;
14         contTurn.enabled = !snap;
15         UpdateTurnText();
```

```
16      }
17      public void SetTurnSpeed(float speed){
18          int speedInt = Mathf.RoundToInt(speed);
19          snapTurn.turnAmount = speedInt * 15f;
20          contTurn.turnSpeed = speedInt * 30;
21          UpdateTurnText();
22      }
23      void UpdateTurnText(){
24          if(snapTurn.enabled) turnTMP.SetText(Mathf.
                RoundToInt(snapTurn.turnAmount) + "°");
25          else turnTMP.SetText(Mathf.RoundToInt(contTurn.
                turnSpeed) + "°/s");
26      }
27      public void SetMoveSpeed(float speed){
28          move.moveSpeed = speed / 100f;
29          UpdateMoveText();
30      }
31      void UpdateMoveText(){
32          moveTMP.SetText(Mathf.RoundToInt(move.moveSpeed*100)
                + "%");
33      }
34      public void SetComfort(int comfortType){
35          switch(comfortType){
36              case 2:
37                  break;
38              case 1:
39                  break;
40              case 0:
41              default:
42                  break;
43          }
44      }
45  }
```

Výpis kódu 4.2 MenuSettings.cs

```
1   public class MenuController : MonoBehaviour
2   {
3       [SerializeField] InputActionProperty leftShowButton;
4       [SerializeField] InputActionProperty rightShowButton;
5       [SerializeField] GameObject menuObj;
6       [SerializeField] Transform headset;
7       [SerializeField] float lerpSpeed;
8       bool rotate = false;
9       void Update() {
10          if(menuObj.activeSelf){
11              transform.position = Vector3.Lerp(transform.
                    position, new Vector3(headset.position.x,
                    transform.position.y, headset.position.z),
                    Time.deltaTime * lerpSpeed);
```

```
12              if (Mathf.Abs(transform.rotation.eulerAngles.y -
                    headset.rotation.eulerAngles.y) > 30) rotate
                    = true;
13              else if (Mathf.Abs(transform.rotation.
                    eulerAngles.y - headset.rotation.eulerAngles.
                    y) < 1) rotate = false;
14              if(rotate) transform.rotation = Quaternion.Lerp(
                    transform.rotation, Quaternion.Euler(0f,
                    headset.rotation.eulerAngles.y, 0f), Time.
                    deltaTime * lerpSpeed);
15          }
16          if (leftShowButton.action.WasPressedThisFrame() ||
                rightShowButton.action.WasPressedThisFrame()){
17              menuObj.SetActive(!menuObj.activeSelf);
18              if (menuObj.activeSelf){
19                  transform.rotation = Quaternion.Euler(0f,
                        headset.rotation.eulerAngles.y, 0f);
20                  transform.position = new Vector3(headset.
                        position.x, transform.position.y, headset
                        .position.z);
21              }
22          }
23      }
24  }
```

Výpis kódu 4.3 MenuController.cs

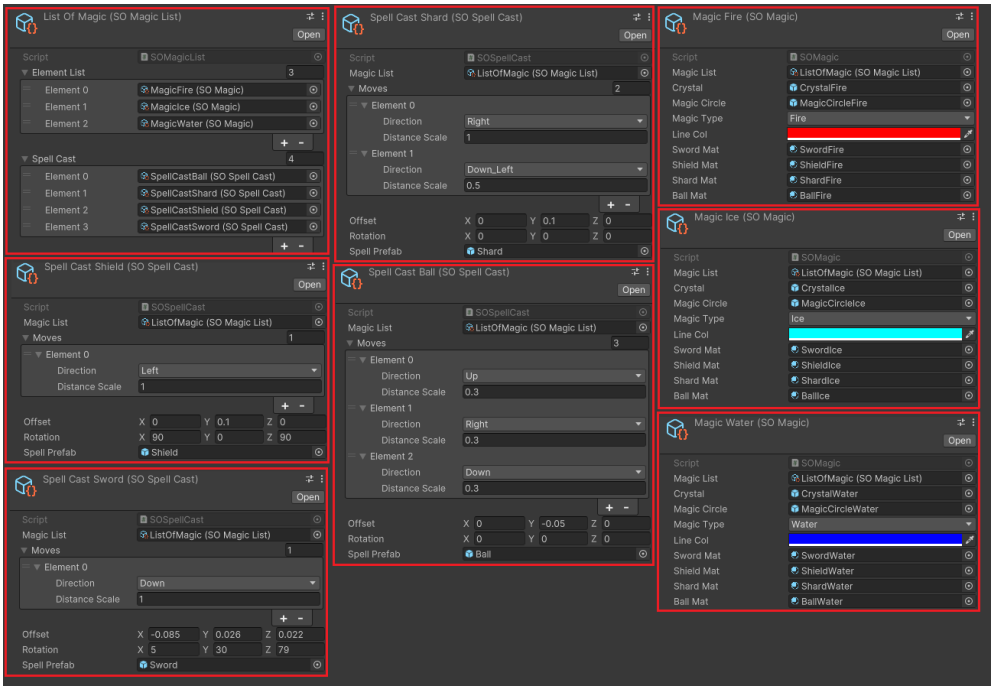
4.2 Implementace magie

Na každou ruku jsem přidal *Sphere Collider* a *Rigidbody* s zapnutým nastavení *Is Kinematic*, aby fungovaly kolize bez počítání fyziky (gravitace, odrazy, …). Pro ukládání všech kouzel a elementů jsem vytvořil SO (scriptable object) *SOMagicList*, ze kterého jsem vytvořil instanci *ListOfMagic*, ve které jsou dva listy (kolekce) SO pro elementy a kouzla.

Elementy mají všechny potřebné informace v instancích SO *SOMagic*. Mezi tyto informace patří: prefab krystalu a magického kruhu pro přepínání, barva čáry (při sesílání kouzel), enum proměnná identifikující element, a věci (Materiály) potřebné pro jednotlivé typy kouzel, …Texturu pro magické kruhy jsem vytvořil s pomocí textury [13] zdarma dostupné na internetu. Materiály použité pro kouzla jsem vytvořil s pomocí *FX Shader Masks* [14], které jsou zdarma stihnutelné na AssetStore.

Kouzla jsou uloženy taktéž v instancích SO *SOSpellCast*, kde jsou informace o: prefabu kouzla s offsetem a rotací a definicí pohybů pro seslání kouzla strukturovaného v listu (kolekci) skládajícího se z enum směru a poměrové hodnoty délky. Důvod proč jsem použil SO, je že ukládá data staticky mimo scénu a umožňuje jednoduché editování instancí v Inspektoru. Tento systém

umožňuje jednoduché a rychlé přidávání či změny kouzel a elementů bez nutnosti zásahu do kódu.



Obrázek 4.5 Úložný systém pro kouzla a elementy (SO)

4.2.1 Zpracování vstupů

Celé zpracování vstupů se odehrává ve scriptech *MagicController* a *CrystalController*. Oba tyto scripty jsem vložil do každé ruky a nastavil jejich proměnné a reference dle obrázku 4.7. *MagicController* obsahuje vyhodnocování vstupů a třídu *LineDraw* sloužící k vyhodnocování sesílání kouzel- *CrystalController* obstarává prohazování krystalů, vrací hodnotu aktivního krystalu a další pomocné funkce související s krystaly a jejich prohazováním. *MagicController* řeší zpracování vstupů pomoci dvou stavů reprezentovaných číslem, kde každý z nich ovlivňuje jeden ze dvou vstupů (Trigger a Grip). Vyhodnocovaní vstupů probíhá ve funkci *void FixedUpdate()*, kterou Unity každou sekundu zavolá vždy podle fixní hodnoty (100). Nejdříve se přečtou vstupy (jestli jsou zmáčknuté nebo ne), které poté pro vstup Trigger zavolají funkci *TriggerActive()* nebo *TriggerNotActive()*, to samé platí pro vstup Grip. Následně se v každé z těchto funkcí rozhodne podle momentálního stavu, co se má stát (obrázek 4.6). Díky tomuto rozdělení jde v budoucnu tento systém jednoduše změnit či rozšířit.

Nejdříve je popíšu fáze stavů, ve kterých se oba vstupy chovají stejně:

-3. Vstup nereaguje, při puštění přejde do stavu 0.

-2. Informace o zmáčknutí/puštění vstupu jsou posílány dále do seslaného kouzla.

-1. Vstup je blokován a musí se externě odblokovat změnou na jinou fázi.

0. Defaultní stav v klidovém stavu, při zmáčknutí se přepne do fáze 1 a blokuje druhý stav vstupu přechodem do fáze -1.

Fáze specifické pro vstup s tlačítkem Grip:

1. Detekuje a vyhodnocuje pohyb ruky pro prohození krystalů.

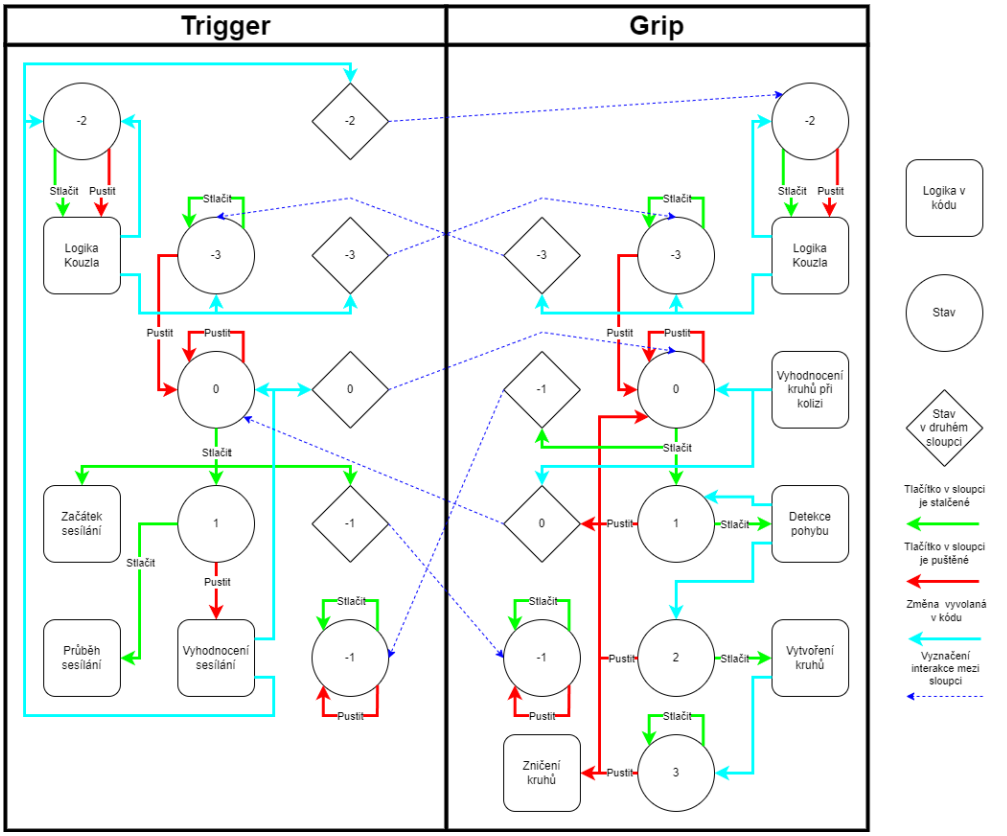
2. Zobrazí magické kruhy pro prohození krystalů.

3. Zničí tyto magické kruhy.

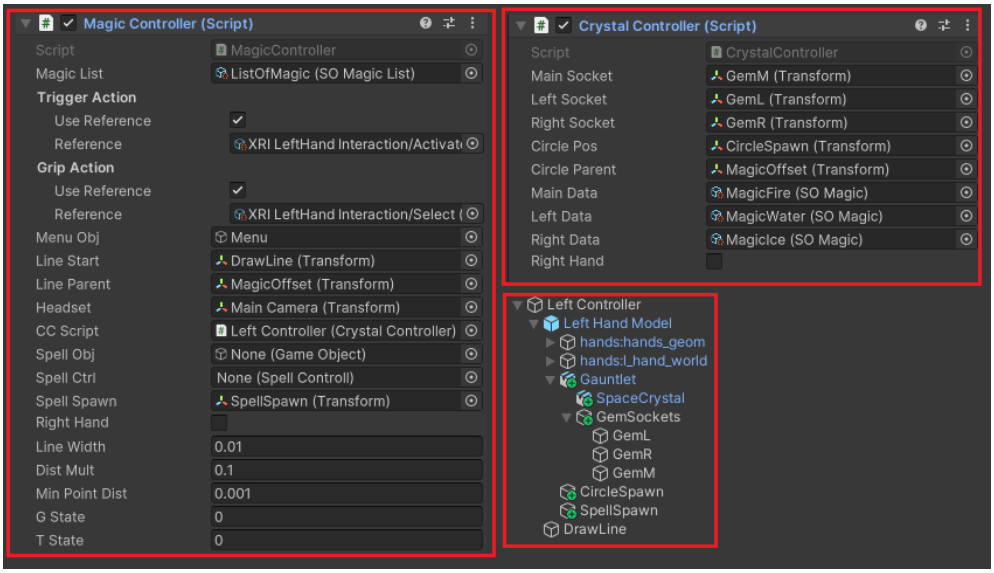
Stavy specifické pro vstup s tlačítkem Trigger:

0. Dodatečně k standardnímu chování spustí inicializaci *LineDraw*

1. Držením vstupu zaznamenává pohyb do *LineDraw* a kreslí čáru ve vzduchu, při puštění vyhodnotí a rozhodne, jestli se sešle kouzlo.



Obrázek 4.6 Stavy a jejich přechody pro Grip a Trigger



Obrázek 4.7 Magic Controll (Script)

4.2.2 Prohození krystalů

Prohození krystalů je provedeno dle postupu popsaného v návrhu (obrázek 3.1). Tento proces je řízený ve scriptu *MagicController*, kde se při zmáčknutí Gripu uloží pozice a rotace ruky. Následně se vypočítává vektor od začátku do momentální pozice ruky, kterým určuji délku rozdílu pozic a rozdílu úhlu začáteční pozice s tímto vektorem. Od délky vektoru 0.05 dle jednotek délky v Unity se začíná vyhodnocovat rozdíl úhlu, pokud větší než 60 ° toto gesto se přeruší a fáze stavu Grip se změní na 3. Ale pokud tento úhel zůstane v 60 ° až do délky 0.2, tak se vytvoří magické kruhy dle barev krystalů. Po prostrčení ruky středem jednoho z magických kruhů se krystaly prohodí a kruhy zmizí, zároveň pokud hráč v jakékoliv části tohoto gesta pustí Grip, tak se vše resetuje a musí začít znova. Všechny potřebné datové a vizualizační funkcionality jsou chytře zabaleny ve 4 funkcích ve scriptu *CrystalController* 4.4 tak, aby je stačilo zavolat v *MagicController* a nebylo třeba nic externě řešit.

- *void DestroyCircles()* – Vytvoří magické kruhy podle prefabů uložených v pravém a levém krystalu na správné pozici.
- *void SpawnCircles()* – Zničí všechny magické kruhy, které vytvořil.
- *void SwapGem(bool right)* – Prohodí krystaly, pokud je hodnota *right* pravdivá, tak aktivní s pravým krystalem, v opačném případě aktivní s levým krystalem.
- *SOMagic ActiveGem()* – Vrátí referenci data o magii aktivního krystalu.

```
public class CrystalController : MonoBehaviour
{
    [SerializeField] Transform mainSocket;
    [SerializeField] Transform leftSocket;
    [SerializeField] Transform rightSocket;
    public Transform CirclePos;
    [SerializeField] Transform CircleParent;
    public SOMagic mainData;
    public SOMagic leftData;
    public SOMagic rightData;
    public bool rightHand;
    GameObject leftMC = null, rightMC = null;
    void Start()
    {
        SpawnCrystal(mainData, mainSocket);
        SpawnCrystal(leftData, leftSocket);
        SpawnCrystal(rightData, rightSocket);
    }
    void SpawnCrystal(SOMagic data, Transform socket){
        if(socket.childCount != 0) Destroy(socket.GetChild
            (0).gameObject);
        Instantiate(data.crystal, socket.position, socket.
            rotation, socket);
    }
    public void SwapGem(bool right){
        DestroyCircles();
        SOMagic tmp = mainData;
        if(right){
            mainData = rightData;
            rightData = tmp;
            SpawnCrystal(mainData, mainSocket);
            SpawnCrystal(rightData, rightSocket);
        } else{
            mainData = leftData;
            leftData = tmp;
            SpawnCrystal(mainData, mainSocket);
            SpawnCrystal(leftData, leftSocket);
        }
    }
    public void SpawnCircles(){
        if (leftMC != null || rightMC != null) return;
        leftMC = Instantiate(leftData.magicCircle, CirclePos
            );
        leftMC.transform.localPosition = new Vector3((
            rightHand ? -1 : 1) * 0.15f, 0, 0);
        leftMC.transform.parent = CircleParent;
        SetMagicCircle(leftMC, false);
        rightMC = Instantiate(rightData.magicCircle,
            CirclePos);
```

```
45          rightMC.transform.localPosition = new Vector3((
                rightHand ? 1 : -1) * 0.15f, 0, 0);
46          rightMC.transform.parent = CircleParent;
47          SetMagicCircle(rightMC, true);
48      }
49      void SetMagicCircle(GameObject circle, bool rGem){
50          MagicCircle tmp = circle.GetComponent<MagicCircle>()
                ;
51          tmp.rightGem = rGem;
52          tmp.hand = transform;
53      }
54      public void DestroyCircles(){
55          if (leftMC != null){
56              Destroy(leftMC);
57              leftMC = null;
58          }
59          if (rightMC != null){
60              Destroy(rightMC);
61              rightMC = null;
62          }
63      }
64
65  }
```

Výpis kódu 4.4 CrystalController.cs

4.2.3 Sesílání kouzel

Téměř celá logika probíhá v třídě *LineDraw* a je ovládána ze scriptu *MagicController* následujících pomocí 3 metod:

- *void DrawStart(Transform transform, Color color)* – Uloží počáteční pozici a nastaví základní informace včetně barvy čáry.
- *void DrawUpdate(Transform transform)* – Ukládá průběžné pozice a zobrazuje je jako spojené body.
- *int DrawEnd(SOMagicList mL)* – Chytře zredukuje počet bodů, přepočítá je na vektory se směrem a porovná je, jestli se neschodují s nějakým definovaným gestem v *SOMagicList*. Při nalezení schody vrátí pořadové číslo kouzla a v opačném případě vrátí hodnotu -1;

Nejdůležitějším procesem je zredukování počtu bodů, jejich přepočítání na směrové vektory a způsob porovnávání s definovanými gesty. Redukce počtu bodů probíhá pomocí dvou v sobě vnořených rekurzí, které postupně projdou všechny body a s zadanou mírou aproximace odstraní přebytečné body. Výpočet využívá výpočtu nejmenší vzdálenosti bodu *x0* od přímky tvořené body *x1, x2* (obrázek 4.8), který jsem nalezl na stránkách aplikace Wolfram [15]. Toto procházení bodů ve funkci *void RecFix(int x1, int x2)* 4.5 začne s indexem

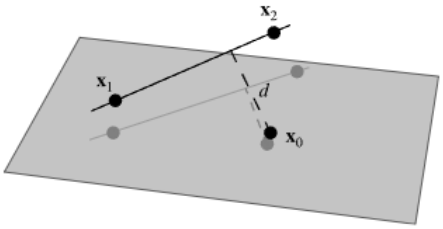
krajních bodů, z nichž vypočítá index bodu, který je zhruba uprostřed mezi nimi. Nejdříve se zkontroluje jestli je vzdálenost bodu *x0* do určité hodnoty od přímky *x1, x2*, pokud není, tak se předpokládá, že tento bod neleží poblíž této přímky a rekurzivně se tato funkce spustí dvakrát po sobě s hodnotami krajních bodů *x1, x0* a *x0, x2*. V opačném případě se druhým typem rekurze *bool RecTest(int from, int to,int x1, int x2, ref float dist)* postupně ověří jestli všechny body v levé *x1, x0* a pravé *x0, x2* straně od bodu *x0* jsou rovné, pokud obě jsou, tak se bod *x2* přidá do vytříděného seznamu bodů. Pokud se hned v levé části stane, že není rovná, tak se vrátí z této rekurze index bodu, který jako první vyčnívá *xt* a následně se spustí klasická rekurze s hodnotami *x1, xt*, *xt, x0* a *x0, x2*. Pokud levá strana je rovná a pravá není, tak se přidá bod *x0* mezi vytříděné body a stejně jako v předchozím případě se vrátí první bod, který vyčnívá *xt* a následně se spustí klasická rekurze s hodnotami *x0, xt* a *xt, x2*. Tento proces se opakuje třikrát po sobě, aby se dosáhlo požadovaného zjednodušení počtu bodů.

Následně se jednotlivé body přepočítají převedením do *struct LDLine*, který vypočítá vše co je potřeba v konstruktoru a umožňuje porovnání s enum *Direction* 4.6. Vstupními informacemi do konstruktoru jsou body *x1,x2*, a pozice headestu, z těchto souřadicí se vypočítá směr této úsečky relativně proti směrovému vektoru, který udává směr od headsetu do středu této úsečky.

Nakonec se postupně porovná směr *LDLine* s definovanými směry gest jednotlivých kouzel pomocí funkce *bool RecSpellTest(int cast, int line)* 4.7. Nejdříve se otestuje jestli jednotlivé indexy ve vstupu funkce nejsou mimo velikost pole, pokud oba jsou, znamená to, že rekurze úspěšně našla shodu a vrátí true. V případě, že jenom jeden z nich je mimo rozsah, tak to není shoda a vrátí false. Pokud ani jeden z nich není mimo velikost svého pole, pokračuje se na porovnání daného indexu *LDLine* s jedním krokem gesta, pokud úsečka není v daném směru, vrátí false. V opačném případě zkusí pokračovat v rekurzi s posunutím pouze indexu *LDLine* a pokud to nevyjde tak zkusí rekurzi s posunutím obou indexů, když to vyjde vrátí true jinak false.

Point-Line Distance--3-Dimensional

Download Wolfram Notebook



Let a line in three dimensions be specified by two points $\mathbf{x}_1 = (x_1, y_1, z_1)$ and $\mathbf{x}_2 = (x_2, y_2, z_2)$ lying on it, so a vector along the line is given by

$$\mathbf{v} = \begin{bmatrix} x_1 + (x_2 - x_1)\,t \\ y_1 + (y_2 - y_1)\,t \\ z_1 + (z_2 - z_1)\,t \end{bmatrix}. \tag{1}$$

The squared distance between a point on the line with parameter t and a point $\mathbf{x}_0 = (x_0, y_0, z_0)$ is therefore

$$d^2 = [(x_1 - x_0) + (x_2 - x_1)\,t]^2 + [(y_1 - y_0) + (y_2 - y_1)\,t]^2 + [(z_1 - z_0) + (z_2 - z_1)\,t]^2. \tag{2}$$

To minimize the distance, set $d\left(d^2\right)/d\,t = 0$ and solve for t to obtain

$$t = -\frac{(\mathbf{x}_1 - \mathbf{x}_0)\cdot(\mathbf{x}_2 - \mathbf{x}_1)}{|\mathbf{x}_2 - \mathbf{x}_1|^2}, \tag{3}$$

where $\cdot$ denotes the dot product. The minimum distance can then be found by plugging t back into (2) to obtain

$$d^2 = (x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2 + 2\,t\,[(x_2 - x_1)(x_1 - x_0) + (y_2 - y_1)(y_1 - y_0) + (z_2 - z_1)(z_1 - z_0)] + t^2\left[(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2\right] \tag{4}$$

$$= |\mathbf{x}_1 - \mathbf{x}_0|^2 - 2\,\frac{[(\mathbf{x}_1 - \mathbf{x}_0)\cdot(\mathbf{x}_2 - \mathbf{x}_1)]^2}{|\mathbf{x}_2 - \mathbf{x}_1|^2} + \frac{[(\mathbf{x}_1 - \mathbf{x}_0)\cdot(\mathbf{x}_2 - \mathbf{x}_1)]^2}{|\mathbf{x}_2 - \mathbf{x}_1|^2} \tag{5}$$

$$= \frac{|\mathbf{x}_1 - \mathbf{x}_0|^2\,|\mathbf{x}_2 - \mathbf{x}_1|^2 - [(\mathbf{x}_1 - \mathbf{x}_0)\cdot(\mathbf{x}_2 - \mathbf{x}_1)]^2}{|\mathbf{x}_2 - \mathbf{x}_1|^2}. \tag{6}$$

Using the vector quadruple product

$$(\mathbf{A}\times\mathbf{B})^2 = \mathbf{A}^2\,\mathbf{B}^2 - (\mathbf{A}\cdot\mathbf{B})^2 \tag{7}$$

where $\times$ denotes the cross product then gives

$$d^2 = \frac{|(\mathbf{x}_2 - \mathbf{x}_1)\times(\mathbf{x}_1 - \mathbf{x}_0)|^2}{|\mathbf{x}_2 - \mathbf{x}_1|^2}, \tag{8}$$

and taking the square root results in the beautiful formula

$$d = \frac{|(\mathbf{x}_2 - \mathbf{x}_1)\times(\mathbf{x}_1 - \mathbf{x}_0)|}{|\mathbf{x}_2 - \mathbf{x}_1|} \tag{9}$$

$$= \frac{|(\mathbf{x}_0 - \mathbf{x}_1)\times(\mathbf{x}_0 - \mathbf{x}_2)|}{|\mathbf{x}_2 - \mathbf{x}_1|} \tag{10}$$

. (11)

Here, the numerator is simply twice the area of the triangle formed by points $\mathbf{x}_0, \mathbf{x}_1$, and $\mathbf{x}_2$, and the denominator is the length of one of the bases of the triangle, which follows since, from the usual triangle area formula, $\Delta = b\,d/2$.

Obrázek 4.8 Výpočet vzdálenosti bodu od přímky [15]

```
void FixPoints(){
    sortedPoints.Clear();
    sortedPoints.Add(0);
    int len = points.Count()-1;
    RecFix(0,len);
    LDPoints newPoints = new ();
    Debug.Log(len + " -> " + sortedPoints.Count() + " :");
    foreach (int i in sortedPoints)
        newPoints.Add(points[i]);
    points.Clear();
    points = newPoints;
}
void RecFix(int x1, int x2){ //4 6
    int x0 = x1 + (x2 - x1) / 2;
    if (x2 - x1 < 2){
        if (x2 - x1 >= 1) sortedPoints.Add(x2);
        return;
    }
    float pointDistance = DistPtoL(points[x0].Pos, points[x1
       ].Pos, points[x2].Pos);
    float LineLength = Vector3.Distance(points[x1].Pos,
       points[x2].Pos) * distMult;
    if(pointDistance < LineLength){
        //bod je rovny
        if(RecTest(x1,x0,x1,x2,ref LineLength)){
            //prvni cast je rovna
            if(RecTest(x0,x2,x1,x2,ref LineLength)){
                //obe casti jsou rovne
                sortedPoints.Add(x2);
            } else{
                //prvni cast je rovna a druha ne -> bod
                   zlomu je dalsi bod:  x0 ... xt ... x2;
                sortedPoints.Add(x0);
                int xt = Mathf.RoundToInt(LineLength);
                RecFix(x0, xt);
                RecFix(xt, x2);
            }
        } else{
            //prvni neni rovna -> rozdeli na 3 x1 ... xt ...
                x0 ... x2
            int xt = Mathf.RoundToInt(LineLength);
            RecFix(x1, xt);
            RecFix(xt, x0);
            RecFix(x0, x2);
        }
    } else{
        //bod neni rovny rozdeleni na x1 ... x0 ... x2
        RecFix(x1, x0);
        RecFix(x0, x2);
```

```
46      }
47  }
48  bool RecTest(int from, int to,int x1, int x2, ref float dist
        ){
49      if (to - from < 3) return true;
50      int x0 = from + (to - from) / 2;
51      float dist3X = DistPtoL(points[x0].Pos, points[x1].Pos,
            points[x2].Pos);
52      if (dist3X < dist)
53          return RecTest(from, x0, x1, x2, ref dist) &&
                RecTest(x0, to, x1, x2, ref dist);
54      dist = x0;
55      return false;
56  }
57  float DistPtoL(Vector3 X0, Vector3 X1, Vector3 X2){ return
        Vector3.Cross(X2 - X1, X1 - X0).magnitude / (X2 - X1).
        magnitude; }
```

Výpis kódu 4.5 FixPoints()

```
1   public LDLine(LDPoint x1, LDPoint x2, Vector3 headset){
2       Start = x1.Pos;
3       Vector3 vector = x2.Pos-x1.Pos;
4       Dist = vector.magnitude;
5       Rot = x2.Rot - x1.Rot;
6       vector = vector.normalized;
7       Vector3 localDir = (x1.Pos + x2.Pos) / 2 + headset;
8       Move = new Vector3(Vector3.Angle(vector, new Vector3(-
            localDir.z, 0, localDir.x).normalized) - 90, Vector3.
            Angle(vector, down) - 90, Vector3.Angle(vector, new
            Vector3(-localDir.x, 0, -localDir.x).normalized) -
            90);
9       float absX = Mathf.Abs(Move.x);
10      float absY = Mathf.Abs(Move.y);
11      float absZ = Mathf.Abs(Move.z);
12      XDir = absX < minAngle ? Direction.None : (Move.x < 0 ?
            Direction.Left : Direction.Right);
13      YDir = absY < minAngle ? Direction.None : (Move.y < 0 ?
            Direction.Down : Direction.Up);
14      ZDir = absZ < minAngle ? Direction.None : (Move.z < 0 ?
            Direction.Back : Direction.Forward);
15      MixDir = XDir | YDir | ZDir;
16      if (absX >= absY && absX >= absZ)
17          MainDir = XDir;
18      else if (absY >= absZ)
19          MainDir = YDir;
20      else
21          MainDir = ZDir;
22  }
```

Výpis kódu 4.6 Konstructor struct LDLine

```
bool RecSpellTest(int cast, int line){
    if (cast == castMaxRST && line == lineMaxRST) return
       true;   //oboje doslo na konec
    if (cast == castMaxRST || line == lineMaxRST) return
       false;   //jedno doslo a dale to nejde
    if (lines.data[line].TestDir(tmpRST.moves[cast].
       GetDirection(rightHand)))
        if (RecSpellTest(cast, line + 1) ||
            RecSpellTest(cast + 1, line + 1))
            return true;
    return false;
}
```

Výpis kódu 4.7 bool RecSpellTest(int cast, int line)

4.2.4 Ovládání kouzel

Každý prefab kouzla musí mít na sobě potomka abstraktní třídy *SpellControll* 4.8 (jednotlivá kouzla na sobě mají scripty, které dědí tohoto scriptu). Ten definuje základní metody a proměnné, které pak všichni ostatní musí obsahovat, čímž umožňuje jednotný interface ke všem kouzlům. Hlavní 4 metody, které obsahuje už v základu jsou:

- *void SetData(SOMagic magicType, MagicController magicController)* – Uloží obě proměné a zavolá abstraktní metodu *SetMaterial()*, která nastaví texturu prefabům.
- *bool Trigger(bool pressed)* – Je neustále volána s momentálním stavem vstupu, ten filtruje a volá abstraktní metodu *bool InputTrigger(bool pressed)* pouze jednou při změně stavu.
- *bool Grip(bool pressed)* – Je neustále volána s momentálním stavem vstupu, ten filtruje a volá abstraktní metodu *bool InputGrip(bool pressed)* pouze jednou při změně stavu.
- *void OnCollisionEnter(Collision other)* – Při kolizi zavolá abstraktní metodu *void MyColl(Collision other)* a pokud detekovaný objekt obsahuje script *SpellInteract*, tak na něm zavolá metodu *.SpellContact(magicType)*.

Bylo potřeba abych vytvořil abstraktní třídu *SpellInteract* 4.9, která zajišťuje jednotný interface na kolizi s kouzly. Z ní jsem poté pro ukázku vytvořil dva potomky *SpellInteractDestroy*, který po kontaktu s jakýmkoliv kouzlem vymaže sám sebe, a *SpellInteractTorch*, který zastupuje ukázku možnosti komplexnější interakce tím, že zhasne pochodeň po použití ledu nebo vody a zapálí ji při použití ohnivého elementu.

```
abstract public class SpellControll : MonoBehaviour
{
    public MeshRenderer model;
    public SOMagic magicType;
    protected MagicController magicController;
    protected int state = 0;
    public void SetData(SOMagic magicType, MagicController
        magicController){
        this.magicType = magicType;
        this.magicController = magicController;
        SetMaterial();
    }
    bool trigger = false, grip = false;
    public bool Trigger(bool pressed){
        if (pressed == trigger) return false;
        return InputTrigger(trigger = pressed);
    }
    public bool Grip(bool pressed){
        if (pressed == grip) return false;
        return InputGrip(grip = pressed);
    }
    void OnCollisionEnter(Collision other){
        SpellInteract tmp = other.gameObject.GetComponent<
            SpellInteract>();
        MyColl(other);
        if (tmp != null){
            tmp.SpellContact(magicType);
        }
    }
    abstract protected void MyColl(Collision other);
    abstract protected void SetMaterial();
    abstract protected bool InputTrigger(bool pressed);
    abstract protected bool InputGrip(bool pressed);
}
```

Výpis kódu 4.8 SpellControll.cs

```
abstract public class SpellInteract : MonoBehaviour
{
    abstract public void SpellContact(SOMagic magicType);
}
```

Výpis kódu 4.9 SpellInteract.cs

Kapitola 5
Hodnocení

Nejdříve ohodnotím kvalitativní a poté kvantitativní analýzou implementaci, kterou budu doplňovat komentáři, jak lze dané části v budoucnu vylepšit nebo proč jsem to neudělal. Většina testování byla provedeno na studentech, kteří už aspoň jednou použili VR.

Co se povedlo:

- Nápad na kombinaci typů kouzel s elementy umožňuje rychlé naučení velkého množství kouzel.
- Realizace funkčního herního prostředí a schopnost kouzlit kouzla, která fungují tak jak mají.
- Díky dobře vytvořenému OO návrhu je možné přidávat a měnit jednotlivé herní prvky (ovládání/kouzla/krystaly/objekty).
- Přehledné rozhraní nastavení a jednoduchý design.
- Rychlý a intuitivní způsob přepínání mezi elementy (krystaly).
- Plynulý a srozumitelný způsob sesílání kouzel s vizualizaci pohybu.

Chce doladit:

- Grafika nebyla dokončená. (Lze vyřešit věnováním více času, nebylo to provedeno z důvodu omezeného zaměření této bakalářské práce a nedostatku času.)
- Přesnost detekce gest. (Lze vylepšit algoritmus pro porovnávání provedeného gesta s kouzlem z first-fit na best-fit systém, nebylo opraveno z důvodu nedostatku času.)
- Výpočet kinetické energie hodu koule občas zlobí a chová se jak chce. (Je potřeba nalézt lepší způsob, nebylo opraveno z důvodu nedostatku času.)
- Chybí in-game návod, který by naučil hráče všechny potřebné ovládací prvky. (Nebylo uděláno z důvodu nedostatku času.)

Data z testování:

- Celkem bylo implementovány 4 různé gesta kouzel a 3 krystaly (elementy), což dává dohromady 12 kouzel.
- Testování přesnosti detekce gest proběhlo na 5 účastnících, s výslednou úspěšností detekce gesta 73 %, v 22 % se vytvořilo jiné gesto a v 5 % se nestalo nic.
- Uživatelé potřebovali průměrně 4 minuty k naučení se všech gest.
- Systém běžel stabilně nad 90 FPS bez sekání na 1 testovací sestavě.

Kapitola 6

Závěr

Cílem této práce bylo vytvoření mechaniky pro sesílání minimálně 3 kouzel pomocí gest s pomocí analýzy několika vybraných VR her. V této práci se podařilo splnit oba tyto cíle a vytvořit funkční prototyp systému pro sesílání kouzel pomocí gest a elemtnálních krystalů. Prototyp je ve formě ukázkového dema, které má účel ukázat téměř neomezené možnosti, které tento systém nabízí. Ačkoliv vývoj dosáhl vytyčených cílů a je k dispozici funkční demo ukazující základní funkce, tak se jedná jen o začátek. Vytvořený systém má mnoho způsobů, kterými se může vylepšit a rozšířit podrobněji popsaných v návrhu, na které bohužel nezbyl čas. Největší škoda je, že jsem nestihnul plně doprogramovat všechny kouzla a vyřešit všechny problémy, které se vyskytují v tomto prototypu. Mimo to v implementaci chyběla spousta dobrovolných aspektů, jako jsou zvuky, paricle efekety, animace, lepší modely a další, které by tohoto dema udělali více celistvý zážitek. Určitě plánuji v tomto projektu nadále pokračovat a vylepšovat ho, jelikož pevně věřím v jeho možnosti.

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-BlNRyH5uqPUe1/view.

13. QUEST5. *Magic Circle* [online]. [cit. 2024-04-15]. Dostupné z: https://forum.arcaneodyssey.dev/t/magic-circles-public-use/30510.

14. 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.

15. RESEARCH, Wolfram. *Point-Line Distance–3-Dimensional* [online]. [cit. 2024-04-15]. Dostupné z: https://mathworld.wolfram.com/Point-LineDistance3-Dimensional.html.

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