

ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA STROJNÍ

Studijní program: B0715P270013 – Strojírenství
Studijní specializace: Průmyslové inženýrství a management

BAKALÁŘSKÁ PRÁCE

**Testování orientace ve virtuálním prostoru pomocí různých
možností ovládní**

Autor: Vojtěch Jezl
Vedoucí práce: Doc. Ing. Petr HOŘEJŠÍ, CSc.

Akademický rok 2022/2023

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta strojní

Akademický rok: 2022/2023

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Vojtěch JEZL**
Osobní číslo: **S20B0148P**
Studijní program: **B0715A270013 Strojní inženýrství**
Specializace: **Průmyslové inženýrství a management**
Téma práce: **Testování orientace ve virtuálním prostoru pomocí různých možností ovládní**
Zadávající katedra: **Katedra průmyslového inženýrství a managementu**

Zásady pro vypracování

1. Úvod
2. Analýza současného stavu – Složitost bludiště, orientace ve virtuálním prostředí
3. Tvorba simulátorů ve virtuální realitě
4. Ověření simulátorů
5. Závěr

Rozsah bakalářské práce: **30 – 40 stran**
Rozsah grafických prací: **0**
Forma zpracování bakalářské práce: **tištěná**

Seznam doporučené literatury:

1. OKITA, Alex. *Learning C# Programming with Unity 3D*. Second edition. Routledge, 2019. ISBN 1138336815.
2. SUNG, Kelvin, SMITH, Gregory. *Basic Math for Game Development with Unity 3D: A Beginner's Guide to Mathematical Foundations*. Apress, 2019. ISBN 978-1484254424.
3. LINOWES, Jonathan. *Unity Virtual Reality Projects: Learn Virtual Reality by developing more than 10 engaging projects with Unity 2018*. 2nd Edition. Packt Publishing, 2018. ISBN 978-1788478809.
4. LaVALLE, Steven. M. *Virtual Reality*. Cambridge University Press, 2020. dostupné online na <http://lavalle.pl/vr/>
5. *Oficiální Unity3D návody dostupné na <https://learn.unity.com/>*

Vedoucí bakalářské práce: **Doc. Ing. Petr Hořejší, Ph.D.**
Katedra průmyslového inženýrství a managementu

Konzultant bakalářské práce: **Ing. Miroslav Bednář**
Katedra průmyslového inženýrství a managementu

Datum zadání bakalářské práce: **19. září 2022**
Termín odevzdání bakalářské práce: **26. května 2023**

L.S.

Doc. Ing. Vladimír Duchek, Ph.D.
děkan

Doc. Ing. Michal Šimon, Ph.D.
vedoucí katedry

Prohlášení o autorství

Předkládám tímto k posouzení a obhajobě bakalářskou práci zpracovanou na závěr studia na Fakultě strojní Západočeské univerzity v Plzni.

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této bakalářské práce.

V Plzni dne:

.....

podpis autora

ANOTAČNÍ LIST BAKALÁŘSKÉ PRÁCE

AUTOR	Příjmení Jezl	Jméno Vojtěch		
STUDIJNÍ PROGRAM	B0715A270013 Strojní inženýrství			
VEDOUCÍ PRÁCE	Příjmení (včetně titulů) Doc. Ing. Hořejší, CSc.	Jméno Petr		
PRACOVISŤE	ZČU – FST – KPV			
DRUH PRÁCE	DIPLOMOVÁ	BAKALÁŘSKÁ		Nehodící se škrtněte
NÁZEV PRÁCE	Testování orientace ve virtuálním prostoru pomocí různých možností ovládání			

FAKULTA	Strojní	KATEDRA	KPV	ROK ODEVZD.	2023
----------------	---------	----------------	-----	--------------------	------

POČET STRAN (A4 a ekvivalentů A4)

CELKEM	57	TEXTOVÁ ČÁST	47	GRAFICKÁ ČÁST	10
---------------	----	---------------------	----	----------------------	----

<p style="text-align: center;">STRUČNÝ POPIS (MAX 10 ŘÁDEK)</p> <p>ZAMĚŘENÍ, TÉMA, CÍL POZNATKY A PŘÍNOSY</p>	<p>Bakalářská práce zpracovává tvorbu 9 simulačních prostředí v Unity, pro 3 různé druhy ovládání ve virtuální realitě. Zaznamenává průběh testování s vytvořenou metodikou na vzorku 15 lidí. Dále je zahrnuta diskuse výsledků v podobě dat z dotazníku a naměřených časech průchodu prostředím. Ovládání ovladači bylo ohodnoceno, jako nepřívětivější a Cybershoes, jako nejrychlejší. Posledním bodem je doporučení pro další výzkum.</p>
<p style="text-align: center;">KLÍČOVÁ SLOVA</p> <p style="text-align: center;">ZPRAVIDLA JEDNOSLOVNÉ POJMY, KTERÉ VYSTIHUJÍ PODSTATU PRÁCE</p>	<p style="text-align: center;">virtuální realita, orientace, unity, Cybershoes, hand-tracking, cybersickness, bludiště, testování, 3D</p>

SUMMARY OF BACHELOR SHEET

AUTHOR	Surname Jezl	Name Vojtěch	
STUDY PROGRAMME	B0715A270013 Mechanical Engineering		
SUPERVISOR	Surname (Inclusive of Degrees) Doc. Ing. Hořejší, CSc.	Name Petr	
INSTITUTION	ZČU – FST – KPV		
TYPE OF WORK	DIPLOMA	BACHELOR	Delete when not applicable
TITLE OF THE WORK	Testing Orientation in Virtual Space Using Different Control Options		

FACULTY	Mechanical Engineering	DEPARTMENT	KPV	SUBMITTED IN	2023
----------------	------------------------	-------------------	-----	---------------------	------

NUMBER OF PAGES (A4 and eq. A4)

TOTALLY	57	TEXT PART	47	GRAPHICAL PART	10
----------------	----	------------------	----	-----------------------	----

BRIEF DESCRIPTION TOPIC, GOAL, RESULTS AND CONTRIBUTIONS	<p>The bachelor thesis deals with the creation of 9 simulation environments in Unity, for 3 different types of control in virtual reality. It records the progress of testing with the developed methodology on a sample of 15 people. Also included is a discussion of the results in the form of questionnaire data and measured times through the environment. Controlling with controllers was rated, as the least friendly and Cybershoes, as the fastest. The last point is a recommendation for further research.</p>
KEY WORDS	virtual reality, orientation, unity, Cybershoes, hand-tracking, cybersickness, maze, testing, 3D

Obsah

Seznam obrázků	10
Seznam tabulek	12
Seznam příloh.....	12
1 Úvod.....	13
2 Prostorová orientace.....	14
2.1 Prostor virtuální reality.....	14
2.2 Rozdíly ve virtuální realitě a reálném světě	15
2.2.1 Vzdálenost.....	15
2.2.2 Schopnost navigace	16
2.3 Problémy spojené s virtuální realitou	16
2.3.1 Cybersickness.....	16
2.3.2 Příčiny a řešení	17
3 Ovládání	18
3.1 Meta Quest 2.....	18
3.2 Hand tracking	18
3.3 Ovladače	20
3.4 Cybershoes.....	21
4 Bludiště.....	22
4.1 Labyrint a bludiště	22
4.1.1 Labyrint	22
4.1.2 Bludiště.....	22
4.2 Definice bludiště.....	23
4.3 Základní prvky.....	23
4.4 Typy a řazení podle vzhladu.....	24
4.4.1 Rozdělení dle typu mozaiky	24
4.4.2 Rozdělení dle typu dimenze	25
4.4.3 Rozdělení dle typu cest	25
4.4.4 Další rozdělení.....	26
4.5 Shrnutí	26
5 Tvorba modelu bludiště.....	27
5.1 Unity 3D	27
5.2 Generátor bludiště.....	27
5.2.1 Porovnání generátorů bludiště.....	27

5.2.2	Umbra Maze Magician.....	28
5.2.3	Seznámení s UMM.....	28
6	Klasifikace bludiště.....	31
6.1	Klasifikace dle teorie grafů.....	31
6.1.1	Dijkstrův algoritmus.....	31
6.1.2	A* a B*.....	31
6.1.3	Další algoritmy.....	32
6.2	Klasifikace na základě geometrie.....	32
6.3	Stanovení metody klasifikace.....	34
6.4	Parametry.....	34
7	Tvorba simulátorů ve virtuální realitě.....	35
7.1	Vytvoření projektu.....	35
7.1.1	Seznámení s uživatelským prostředím unity.....	36
7.1.2	Nastavení projektu.....	37
7.1.3	Instalace Assetu.....	39
7.2	Tvorba simulátorů.....	41
7.2.1	Vytvoření pohledu bludiště.....	41
7.2.2	Hledání správné cesty.....	45
7.2.3	Generování bludišť.....	48
7.2.4	Skripty.....	50
7.2.5	VR funkčnost.....	56
7.3	Shrnutí.....	57
8	Ověření simulátorů.....	59
8.1	Metodika testování.....	59
8.1.1	Úvodní část.....	59
8.1.2	Testovací část.....	59
8.1.3	Závěrečná část.....	60
8.1.4	Testovací skupina.....	60
8.2	Průběh měření.....	61
8.3	Výsledky měření.....	61
8.4	Vyhodnocení.....	69
9	Závěr.....	70
10	Seznam použité literatury.....	71
	PŘÍLOHA č. 1.....	i

Seznam obrázků

Obrázek 1 Egocentric	14
Obrázek 2 Allocentric	14
Obrázek 3 porovnání FOV VR headsetů [3].....	15
Obrázek 4 VR headset [10]	18
Obrázek 5 Hand tracking ve VR [12].....	19
Obrázek 6 Pinch [15].....	19
Obrázek 7 LED diody na Meta Quest 2 [16]	20
Obrázek 8 Umístění tlačítek [18]	20
Obrázek 9 Cybershoes [20].....	21
Obrázek 10 Židle s kobercem [20].....	21
Obrázek 11 Labyrint [21]	22
Obrázek 12 Bludiště [21]	22
Obrázek 13 Druhy buněk	23
Obrázek 14 Ortogonální bludiště [24].....	24
Obrázek 15 Omega Bludiště [24].....	24
Obrázek 16 Více dimenzionální bludiště [23].....	25
Obrázek 17 Create Maze Generator	28
Obrázek 18 Gizmo	29
Obrázek 19 Nastavení v UMM	30
Obrázek 20 Dijkstrův algoritmus	31
Obrázek 21 Bludiště 5x5	32
Obrázek 22 Bludiště 5x5 – úseky.....	33
Obrázek 23 Bludiště 5x5 – slepé konce	33
Obrázek 24 Bludiště 5x5 – rozcestníky.....	33
Obrázek 25 Přihlášení	35
Obrázek 26 Zakládání projektu	35
Obrázek 27 Unity Editor	37
Obrázek 28 Unity Editor – přizpůsobený.....	37
Obrázek 29 Build Settings.....	38
Obrázek 30 Project Settings – Player	38
Obrázek 31 Project Settings – Quality	39
Obrázek 32 Project Setting - XR Plug-in	39
Obrázek 33 Package Manager – UMM.....	40
Obrázek 34 Package Manager – OI.....	40

Obrázek 35 Blender – Model regálu	41
Obrázek 36 UV unwrap.....	42
Obrázek 37 UV unwrapnutý model	43
Obrázek 38 Blender – Nodes	43
Obrázek 39 Nastavení exportu FBX	44
Obrázek 40 Textury.....	44
Obrázek 41 FBX model.....	44
Obrázek 42 Stěna	44
Obrázek 43 Upravený vzhled bludiště	45
Obrázek 44 PathFinding Tool – Editor	45
Obrázek 45 Metody hledání cesty.....	46
Obrázek 46 Pozice bloku bludiště.....	46
Obrázek 47 Generování bludiště	48
Obrázek 48 Kategorie II.....	49
Obrázek 49 Kategorie III.....	49
Obrázek 50 Kategorie I	50
Obrázek 51 Start a cíl.....	51
Obrázek 52 SideQuest.....	57
Obrázek 53 Teleport.....	57
Obrázek 54 Pohyb ovladači.....	58
Obrázek 55 Pohyb Cybershoes	58
Obrázek 56 Zkušenosti respondentů	61
Obrázek 57 SUS hodnocení	62
Obrázek 58 Průměrný čas I. Kategorie	63
Obrázek 59 Průměrný čas II. kategorie.....	63
Obrázek 60 Průměrný čas III. Kategorie.....	64

Seznam tabulek

Tabulka 1 Hodnocení generátorů	28
Tabulka 2 Rozdělení dle velikosti	34
Tabulka 3 Bludiště I	49
Tabulka 4 Bludiště II	49
Tabulka 5 Bludiště III	49
Tabulka 6 Parametry testovacích prostředí	50
Tabulka 7 Pořadí dle orientace	61
Tabulka 8 SUS hodnocení	62
Tabulka 9 Ruce – Ovladače	65
Tabulka 10 Ruce – Cyberhsoes	65
Tabulka 11 Ovladače – Cybershoes	66
Tabulka 12 Cybersickness hodnocení	66
Tabulka 13 Průměrné časy ze 3 kategorií	67
Tabulka 14 Pořadí	68

Seznam příloh

VÁZANÉ

Příloha č.1 Dotazník.....	i
---------------------------	---

NEVÁZANÉ

Příloha č. 2 Data	
-------------------	--

1 Úvod

Virtuální realita se v posledních letech těší vysoké popularity a velkého rozmachu ve vývoji aplikací. To je zapříčiněno rostoucí rychlostí vývoje této technologie. Lehce upravitelné, velmi věrohodné virtuální prostředí a intuitivní interakce přináší uživatelům možnost ponořit se do rozmanitého obsahu. V porovnání s reálným světem je vstup do virtuálního prostředí snadnější, s nižšími náklady a v mnoha případech i jedinou možností k dosažení zamýšleného scénáře. Z velké části je VR využívána herním prostředím, nicméně najde se velké uplatnění i v oblasti medicíny, rehabilitace či sportu.

Vzhledem ke zmiňovaným výhodám a rostoucímu zájmu o působení ve virtuálním světě vzniká prostor pro hlubší zkoumání jednotlivých technologií, se kterými se ve VR pracuje. Jednou takovou oblastí je ovládání VR. Různé scénáře ve VR vyžadují odlišné ovládání a tím vzniká otázka, které ovládání je vhodné a které naopak méně. Cílem této práce je tedy přispět k výzkumu vhodnosti konkrétních ovládaní z hlediska rychlého orientování se v prostoru. Tímto prostorem může být například průmyslový sklad. Zaměříme se tak na porovnání konkrétních třech ovládaní, kterými jsou gesty rukou, ovladače a Cybershoes.

Pro testování byla vytvořena simulační prostředí fungující na principu bludišť, které mají znázorňovat již zmiňovaný průmyslový sklad. Tato prostředí jsou rozdělena do třech úrovní náročnosti. Pro každé ovládání pak bude vytvořeno jedno prostředí od každé úrovně. Jako výstup testů je čas průchodu daného „bludiště“ a subjektivní názor respondentů zachycený dotazníkovým průzkumem. Zvyšující se náročnost prostředí by měla mít za důsledek zvyšující se šanci na tvorbu chyb. Předpokladem je tedy, že čas průchodu bude s každou úrovní narůstat, zároveň se prodlouží čas průchodu důsledkem tvorby chyb. S vhodnějším ovládaním by pak člověk měl být schopný tyto chyby omezit, právě díky lepší schopnosti se s daným ovládaním orientovat. Do výstupu by se tak měla projevit schopnost rychlosti průchodu prostorem i schopnost se v něm orientovat. Dále se jednotlivá ovládaní ohodnotí pomocí stupnice použitelnosti systému, tzv. „SUS score“.

Posledním bodem pak bude ověření následujících hypotéz:

Hypotéza (H): Mezi hodnotami průměrných časů ze všech třech kategorií u ovládaní Ruce, Ovladače a Cybershoes existují statisticky významné rozdíly.

Nulová hypotéza (H_0): Mezi hodnotami průměrných časů ze všech třech kategorií u ovládaní Ruce, Ovladače a Cybershoes neexistují statisticky významné rozdíly.

2 Prostorová orientace

Cílem virtuální reality je co nejvěrněji nahradit reálný prostor, ve kterém se člověk pohybuje. Aby bylo možné přesun z reálného do virtuálního světa co nejpřesněji uskutečnit, je třeba porozumět mnoha mechanismům a aspektům, které jsou v běžném životě zcela intuitivní a přirozené.

K nalezení vhodné cesty se člověk opírá o soubor prostorových dovedností, které představují základní předpoklady pro efektivní využití a následné zpracování informací z okolního prostředí. Těmito dovednostmi jsou percepční¹ schopnosti, dovednosti zpracování informací, dříve získané prostorové znalosti a motorické dovednosti, které jsou využívány v závislosti na specifickém druhu pohybu a orientaci člověka.

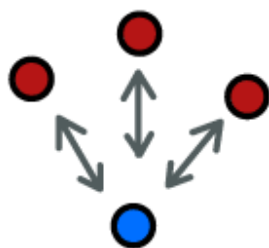
Orientovaný pohyb se dělí do tří hlavních kategorií:

- 1) účelný pohyb k předem známému cíli
- 2) průzkumné cestování s cílem vrátit se do známého výchozího bodu
- 3) pohyb vedoucí k dosažení nového, doposud neznámého cíle

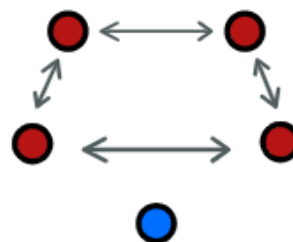
Průvodní experiment této práce se zaměřuje především na třetí kategorii, jehož cílem je mimo jiné zjištění schopnosti orientace při využití různých možností ovládnutí pohybu. [1]

2.1 Prostor virtuální reality

Prostředí virtuální reality můžeme lehce nastavit a upravit podle našich potřebných požadavků, což ho činí ideálním pro účely testování a zkoumání prostorové orientace či vytváření vizuálních iluzí. Proces orientace člověka je založen na dvou systémech: egocentrickém (Obrázek 1) a allocentrickém (Obrázek 2) referenčním systému, které spolupracují při mnoha každodenních aktivitách. Egocentrický systém popisuje relativní pozici a orientaci objektů vůči pozorovateli, zatímco allocentrický systém popisuje vzájemnou polohu objektů v prostoru. Například při hře stolního tenisu musí hráč správně odhadnout pozici rychle letícího míčku a pohybovat se s pálkou tak, aby se míček odpíkl do zamýšleného směru. Tyto interakce jsou založeny na prostorovém vnímání, které však může být ovlivněno virtuálním prostředím a posunutím virtuálních objektů v něm. [2]



Obrázek 1 Egocentric



Obrázek 2 Allocentric

¹ Percepční – Vjemové schopnosti, registrující barvy, zvuk, vůně apod.

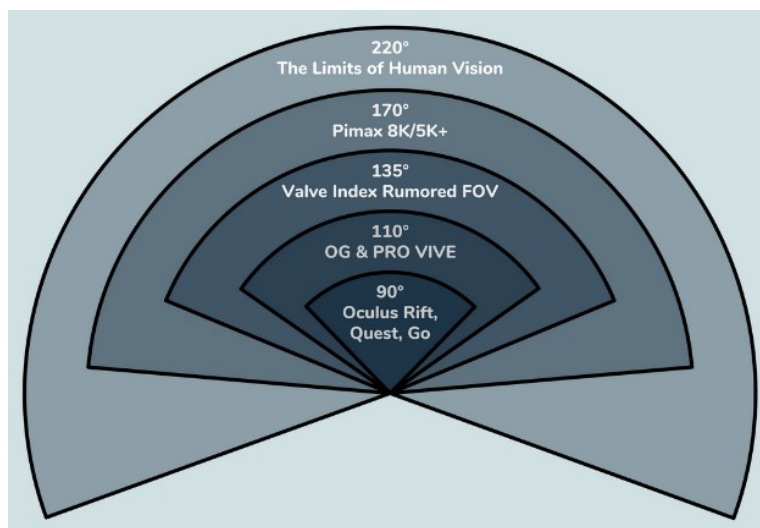
2.2 Rozdíly ve virtuální realitě a reálném světě

Virtuální realita a skutečný svět jsou dvě odlišné prostředí s vlastními charakteristikami a výzvami. Ve skutečném světě mohou lidé vnímat a interagovat s okolím pomocí svých pěti smyslů, které jim umožňují vidět, slyšet, cítit a dotýkat se objektů. Skutečný svět je třírozměrný a objekty v něm mají specifickou velikost, tvar a texturu. Na druhé straně, virtuální realita je prostředí generované počítačem, které je vnímáno pomocí headsetu nebo jiného zařízení. Uživatel není schopen interagovat s virtuálním prostředím stejným způsobem jako ve skutečném světě, a navíc je virtuální svět nebo jeho části často omezen pouze na dvě dimenze, což může vést k nedostatku realismu a detailu. Tyto rozdíly mohou ovlivnit způsob, jakým lidé vnímají a navigují se v rámci virtuálního prostředí a mohou ovlivnit úroveň vnoření a přítomnosti, kterou v něm pociťují.

2.2.1 Vzdálenost

K určení polohy virtuálního objektu ve virtuální prostoru se vizuální systém spoléhá na indikátory vzdálenosti a hloubky, jako je konvergence² a zorné pole (FOV)³. Některé studie ukázaly, že vzdálenosti ve virtuálním světě byly často vnímány kratší, než ve skutečnosti byly. To může být zapříčiněno nedostatečným výkonem zobrazovacího hardwaru, který se potýká s problémy jako je kvalita obrazu, světlo, paralaxa pohybu⁴, rozměr FOV a z povahy počítačové grafiky. Toto podceňování vzdálenosti ve VR bylo často pozorováno z egocentrické perspektivy. Většina studií také zahrnovala testování odhadu vzdálenosti prostřednictvím chůze. Bylo zjištěno, že počítačová grafika může mít vliv na verbální odhady vzdálenosti, ale nemá žádný významný vliv na odhady založené na chůzi.

Za účelem bližšího objasnění rozdílů mezi jednotlivými druhy hardwaru, které ovlivňují subjektivní hodnocení pozorovatele, je uveden v (Obrázek 3), znázorňující rozdíly v zorném poli. Z uvedeného obrázku vyplývá, že i ty nejmodernější VR headsety stále nedokážou plně pokrýt hranice lidského zorného pole.



Obrázek 3 porovnání FOV VR headsetů [3]

² Konvergence – Značí natočení očí, které je potřebné, aby se jejich optické osy sbíhaly při pohledu na předmět, který je blíže než 25 cm. [33]

³ Zorné pole (FOV) – Zorné pole oka je část prostoru, který můžeme vnímat, když upřeme oči na jeden bod. Je definován jak vertikálně, tak horizontálně. [34]

⁴ Paralaxa pohybu – Jedná se o přímý pohyb pozorovatele, při kterém se bližší předměty v jeho zorném poli pohybují rychle, zatímco předměty vzdálenější se pohybují velmi pomalu. [35]

2.2.2 Schopnost navigace

Hledání cesty je denním úkolem, kterému čelíme při běžných aktivitách jako cesta do práce, setkávání s přáteli či plánování dovolené. Tento úkol je natolik důležitou součástí našeho každodenního života, že se stal předmětem zájmu řady výzkumů z různých perspektiv. Současná výzkumná práce v oblasti navigace v kognitivní vědě se snaží porozumět vnitřním znalostem a mentálním procesům, které jsou zapojeny při hledání cesty a s ním souvisejících aktivitách. Tato práce se například zaměřuje na učení se rozložení prostředí v neznámé oblasti, kterou se plánujeme vydat.

Další klíčovou složkou pro vytvoření kognitivní mapy okolí, kterou potřebujeme k přesné prostorové orientaci, je vizuálně-prostorová dovednost. Zrak zde hraje klíčovou roli při poskytování informací o prostorovém okolí. Kromě toho je pro tvorbu vizuálních vodítek při navigaci prostorem nezbytná prostorová paměť. Tyto schopnosti lze přiblížit na příkladu účinnosti evakuace z hořící budovy.

Hledání cesty je definováno jako kognitivní proces, který zahrnuje schopnost naučit se trasu a následně být schopen si tuto trasu vybavit z paměti tak, aby bylo možné se dostat z jednoho místa na druhé, přičemž je třeba neustále odhadovat prostorové vzdálenosti mezi předměty nebo lidmi v okolním prostředí. V této oblasti nebylo prokázáno výrazné zhoršení prostorové orientace ve virtuální realitě v porovnání s reálným světem. [2]

2.3 Problémy spojené s virtuální realitou

Pohyb v prostředí virtuální reality s sebou nese celou řadu výzev a potenciálních problémů, které bychom mohli rozdělit do dvou kategorií. První kategorie jsou fyziologické problémy, které vznikají přímou interakcí lidského těla s virtuální realitou. Tyto problémy mohou zahrnovat například nevolnost, závratě nebo narušení rovnováhy. Druhou kategorií jsou technologické problémy, jako například nedostatek výkonu, nízká kvalita obrazu nebo zpoždění v reakcích na uživatelské chování. Tyto problémy mohou mít negativní dopad na uživatelskou zkušenost a celkovou účinnost prostředí virtuální reality.

2.3.1 Cybersickness

V této práci bude termín "cybersickness" používán jako označení pro virtuální nemoc a nikoli jako kybernetická nemoc, která se v současnosti často používá pro popis různých negativních účinků spojených s technologiemi, jako jsou problémy s dlouhodobým používáním mobilních telefonů a počítačů obecně.

Charakterizace virtuální nemoci pro výzkumníky byla obtížná kvůli mnoha faktorům, jako jsou rozdíly mezi lidmi při přizpůsobení se opakovanému používání, obtížnost měření příznaků a rychle se vyvíjející technologie. Avšak, díky pokroku v technologii virtuální reality, jako jsou zobrazovací, snímací a výpočetní technologie, se tento problém postupně snižuje. Přesto stále existují nežádoucí účinky u citlivějších uživatelů.

Existuje několik teorií, které vysvětlují virtuální nemoc, přičemž nejčastější je teorie neshody smyslů. Tato teorie tvrdí, že nesoulad mezi smyslovým vstupem do vnitřního ucha a očí vede k pocitu kinetózy⁵. V případě virtuální reality nastává tento jev, kdy uživatel pozoruje pohyb v prostředí VR, zatímco jeho tělo zůstává stát na místě.

⁵ Kinetóza – Jinak nazývána nemoc z pohybu, je fyzický stav člověka, kdy vnímání pohybu vestibulárního systému neodpovídá získaným vizuálním vjemům. Tento stav doprovází závrať, vyčerpání nebo nevolnost. [36]

Další teorií je teorie posturální nestability, která tvrdí, že když tělo nedokáže udržet stabilní polohu v prostoru pomocí smyslových vjemů, může to vést k nevolnosti. To se může stát, pokud se v prostředí VR pohybuje příliš rychle nebo pokud jsou předměty v prostředí neustále v pohybu.

Třetí teorie je teorie klidového rámce. Tato teorie tvrdí, že když očekáváme určitý smyslový vjem a dostane se nám jiný, může to vést k nevolnosti. Například, pokud očekáváme, že zůstaneme v klidu na židli, ale v prostředí VR se náhle pohybujeme, může to vést k pocitu nevolnosti. [4] [5]

I když tyto teorie mohou pomoci vysvětlit některé případy cybersickness, stále nejsou úplně jasné a nejsou schopny vysvětlit všechny případy. Někteří lidé jsou více citliví na virtuální realitu než jiní, a také může hrát roli mnoho dalších faktorů, jako jsou individuální rozdíly, předchozí zkušenosti a další.

2.3.2 Příčiny a řešení

V posledních desetiletích se v oblasti virtuální reality objevila problematika tzv. virtuální nemoci. S tímto jevem se zabývá mnoho studií, které se zaměřují na jeho příčiny a důsledky. Tyto studie lze obecně rozdělit do tří kategorií v závislosti na problematice, kterou se zaměřují. Konkrétně se jedná o oblasti hardware, software a jednotlivé uživatele virtuální reality.

Kategorie hardware v kontextu virtuální reality zahrnuje celou řadu komponent, jako je například rozlišení, velikost obrazovky či zpoždění, přičemž toto zpoždění by nemělo přesáhnout hranici 20 milisekund. Pro zajištění lepší rychlosti obrazu je vhodné vylepšit procesor, grafickou kartu a hlavní paměť. Pro zajištění optimálního uživatelského zážitku je doporučena obnovovací frekvence pro každé oko v hodnotě 45 Hz. Tento parametr může být klíčový pro minimalizaci případných projevů virtuální nemoci [6]

Software se v rámci virtuální reality týká aplikace, rychlosti pohybu a celkového vzhledu virtuálního prostředí. Velikost zorného pole (FOV) patří mezi nejvýznamnější faktory ovlivňující výskyt virtuální nemoci. Studie zkoumající tuto problematiku ukazují, že zdvojnásobení velikosti zorného pole se zhruba zdvojnásobí frekvence výskytu a závažnost nevolnosti z virtuální nemoci. Rychlost pohybu začíná být problémová ve vyšších úrovních, kdy uražená vzdálenost ve virtuálním prostoru se výrazně neslučuje s časem, za který uživatel předpokládal tuto vzdálenost urazit. V herním prostředí se tento problém řeší určitými omezeními rychlosti pohybu v podobě ukazatelů výdrže, kdy se hráč může pohybovat vyššími rychlostmi jen po omezenou dobu. Dále je pak vhodné například místo schodů využívat šikmých ploch, které jsou šetrnější k lidským smyslům. [4] [7]

Mezi významné uživatelské faktory ovlivňující výskyt virtuální nemoci patří věk, pohlaví a frekvence používání VR. Avšak, výsledky výzkumů jsou často nepřesné, protože získání dostatečného počtu starších účastníků je stále obtížné. Nicméně, některé studie ukazují, že starší uživatelé mohou vykazovat vyšší výskyt virtuální nemoci v určitých systémech VR. V opakovaných výzkumech se také ukazuje, že ženy jsou více náchylné k projevům virtuální nemoci. Frekvence používání VR také může hrát roli, ale výsledky v této oblasti jsou stále nejednoznačné. [8] [9]

V oblasti prevence virtuální nemoci je důležité zajistit dostatečnou snímkovou frekvenci při používání VR interakčních nástrojů, aby se zabránilo pohybové nevolnosti. Kromě toho mohou vývojáři implementovat volitelná nastavení, která snižují pravděpodobnost vzniku virtuální nemoci. Výzkumy ukazují, že časté používání VR může vést ke zvyklosti a zvýšit toleranci uživatele k nežádoucím účinkům.

3 Ovládání

Pro účely testování, které bude podrobněji popsáno v kapitole 8, budou použity VR headsety⁶ od společností Meta a Valve. Konkrétně se jedná o model Meta Quest 2 od společnosti Meta. Budou využity dvě hlavní funkce tohoto VR headsetu, a to ovládání pomocí hand trackingu⁷, a ovládání pomocí ovladačů. Druhým headsetem je Valve Index od společnosti Valve, který bude spárován s návlakem na boty nazývaným Cybershoes od stejnojmenné společnosti. Pohyb bude ovládán přes samotný hardware Cybershoes a Valve Index headset bude sloužit pouze k zobrazení virtuálního světa pro účastníky testování. Využití druhého headsetu je z důvodu lepší kompatibility Cybershoes, které jsou dostupné k testování.

3.1 Meta Quest 2

Tento typ VR brýlí je nazýván jako „all-in-one“ verze, což znamená, že není potřeba připojení k počítači či externím sensorům pro fungování aplikací. Vše, co je potřeba, je již zabudováno přímo v brýlích.

Meta Quest 2 (Obrázek 4) se výchozím nastavením ovládá pomocí dvou ovladačů, avšak zařízení obsahuje vestavěné senzory pro hand tracking, což znamená, že je možné ovládat zařízení pouze pomocí pohybů rukou. Tato funkce funguje jak v prostředí brýlí, tak i v podporovaných aplikacích. Kromě toho mohou uživatelé připojit další zařízení, jako již zmiňované Cybershoes nebo vestu, pro zvýšení realismu a imerze. [10]



Obrázek 4 VR headset [10]

3.2 Hand tracking

Vstupní metoda, která umožňuje ovládání virtuální reality pomocí rukou, je v poslední době stále častěji používána a výzkum v této oblasti se neustále rozvíjí. Nicméně, stále existují některá omezení, která brání plné implementaci této technologie. Hlavní překážky v této oblasti spočívají v problematických oblastech počítačového vidění a snímání pohybu ruky.

Využití rukou jako ovládacího prvku je již dlouhodobě velmi oblíbené a vyhledávané. Způsobuje to několik faktorů, které dělají z rukou unikátní a bezprecedentní způsob ovládání. Na rozdíl od ovladačů, které mohou být pro nové uživatele komplikované, ruce nabízejí jednodušší a přirozenější způsob ovládání, aniž by bylo nutné se učit ovládat tlačítka a analogové páčky. Používání rukou také eliminuje náklady spojené s externími ovladači, jako jsou náklady na jejich pořízení, provoz či případnou ztrátu. Uživatel může ihned po nasazení VR headsetu okamžitě ovládat virtuální svět, což zlepšuje celkový zážitek ve VR. [11]

⁶ VR Headset – náhlavní zařízení poskytující uživateli propojení s virtuálním světem

⁷ Hand tracking – technologie snímání rukou při ovládání virtuální reality (Obrázek 5)



Obrázek 5 Hand tracking ve VR [12]

Implementace snímání pohybu rukou jako ovládacího prvku v reálném čase čelí určitým technickým překážkám. Ty se týkají náročnosti snímání všech pohybů ruky v reálném čase. Další problémy vycházejí z nedostatku zpětné vazby od virtuálních objektů při manipulaci s nimi pomocí rukou. V důsledku toho se uživatelé mohou setkat s nepřírozeným pocitem při manipulaci s objekty. Navíc může být problematické rozlišit cílené gesto pro ovládání VR a náhodný pohyb rukou, který může nechtěně spustit určité procesy. Ovládání rukou nelze snadno zapnout ani vypnout, což opět může vést k neúmyslným interakcím s virtuálním prostředím. [11]

Gesta k ovládání

Ruce nabízejí téměř neomezenou svobodu pohybu a schopnost vytvářet různá gesta. To otevírá širokou škálu možností, ale příliš mnoho variant může také způsobovat zmatek.

Snímání rukou analyzuje jejich jednotlivé pozice a v reálném čase sleduje polohu klíčových bodů na ruce, jako jsou klouby nebo konečky prstů, při jejich pohybu. Pro ovládání rukama využíváme pozice ruky, podle kterých se řídí kurzor, který se chová jako standardní kurzor ovladače. Pomocí kurzoru můžeme zvýraznit, vybrat nebo kliknout na určitý prvek aplikace. Integrované ruce mohou provádět interakce s předměty pomocí jednoduchých gest rukou, jako je ukazování, svírání, odebírání, posouvání a sevření dlaní. [13]

Při ovládání rukama se nejvíce využívá tzv. „Pinch“ (Obrázek 6), při kterém se dotýká palec s ukazováčkem. Kontakt mezi palcem a ukazováčkem může kompenzovat nedostatek hmatové odezvy, kterou obvykle poskytují jiná vstupní zařízení. Toto gesto funguje se využívá díky své lehké proveditelnosti a snadné zapamatovatelnosti. Navíc se nejedná o běžnou pozici ruky, což minimalizuje riziko nechtěného zachycení gesta systémem. Skládá se ze tří pozic, otevření, uzavírání a zavření a využívá se při veškeré interakci s VR systémem. [14]



Obrázek 6 Pinch [15]

3.3 Ovladače

Ovladače slouží k registraci pohybů rukou a prstů v rámci virtuálního prostředí. Tyto zařízení přenášejí fyzické pohyby rukou do digitálního prostoru a umožňují tak interakci s virtuálním světem.

Většina dostupných ovladačů VR má sadu tlačítek a páčku pro palec, která umožňuje uchopení, stisknutí, házení nebo pohybování virtuálními objekty. Například stisknutí tlačítka na ovladači odpovídá zmáčknutí spouště virtuální zbraně, zatímco použití palcové páčky umožňuje chůzi ve virtuálním světě.

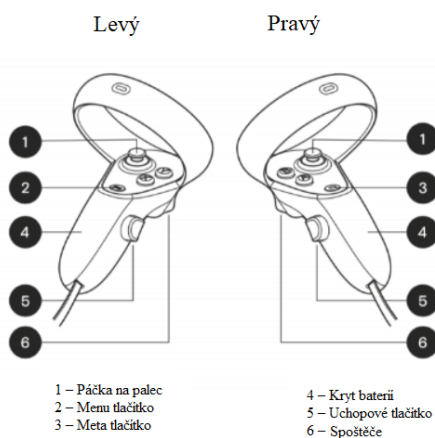
Meta Quest 2 ovladače

Meta Quest 2 používá technologii sledování "zevnitř ven" neboli "inside-out tracking", což znamená, že přední kamery na náhlavní soupravě brýlí musí vždy "vidět" nebo alespoň detekovat část ovladače v jejich zorném poli, aby mohly sledovat jeho pohyby. Každý ovladač je vybaven sadou infračervených LED diod (Obrázek 7) umístěných na prstencích, které jsou detekovány kamerami na náhlavní soupravě. Tyto kamery nepřetržitě pořizují snímky LED diod a na základě těchto snímků sledovací systém trianguluje⁸ polohu ovladačů v prostoru. Tento přístup však může mít určitá omezení, například mohou nastat kolize při sledování, pokud se ovladače překrývají, a nelze efektivně sledovat pohyby rukou za zády.



Obrázek 7 LED diody na Meta Quest 2 [16]

Tyto ovladače jsou vybaveny ergonomickými rukojetěmi a obsahují tlačítka pro ukazováček a prostředníček. Jejich poloha a pohyb jsou snímány prostřednictvím prstenců na ovladačích. V porovnání s ostatními ovladači na trhu poskytují nejdelší výdrž baterie – až 60 hodin aktivního používání na jedno nabití. [17]



Obrázek 8 Umístění tlačítek [18]

⁸ Triangulace je způsob zjišťování souřadnic a vzdáleností. Provádí se trigonometrickým výpočtem. [37]

3.4 Cybershoes

Cybershoes představují nový způsob pohybu v rámci virtuální reality, který se skládá z párů bot, které se nasadí přes běžné boty. Tyto boty obsahují válečky a rotační senzory (Obrázek 9), které sledují pohyby při chůzi a určují směr chůze. To umožňuje uživatelům simulovat pohyb při chůzi ve virtuálním prostředí, aniž by museli fyzicky chodit. Uživatelé místo toho sedí na místě a kloužou chodidly po podlaze, což vytváří pocit, že se pohybují po virtuálním prostoru. Senzory pak určují směr pohybu, takže se uživatelé mohou dívat jedním směrem, zatímco se pohybují jiným. Cybershoes jsou tak inovativním způsobem, jak dosáhnout pohybu v rámci virtuální reality bez nutnosti fyzického pohybu po prostoru. [19]



Obrázek 9 Cybershoes [20]

Cybershoes se prodávají spolu se židlí ("Cyberchair") a kobercem ("Cybercarpet"), aby uživatelé umožnily správně sedět a pohybovat se s optimálním třením (Obrázek 10).



Obrázek 10 Židle s kobercem [20]

Výhody

- 1) Snižuje virtuální nevolnost
Díky simulaci pohybu chůze pomocí klouzání chodidel po podlaze mohou Cybershoes výrazně snížit pocit kinetózy při používání virtuální reality.
- 2) Přesné sledování směru
Uživatelé mohou nezávisle na směru pohybu Cybershoes volně sledovat své okolí, což znamená, že mohou plně vnímat virtuální prostředí, zatímco se pohybují v libovolném směru.
- 3) Sledování aktivity
Uživatelé mohou sledovat svou fyzickou aktivitu při používání Cybershoes a určovat vzdálenost svých virtuálních cest nebo sledovat vydanou energii. [20]

4 Bludiště

Abychom mohli s bludištěm jako nástrojem pro testování orientace ve virtuálním prostoru pracovat, je třeba si nejdříve definovat co považujeme za bludiště, jaké má části a jaký druh využijeme.

4.1 Labyrint a bludiště

Často se stává, že se zaměňuje pojem labyrint a bludiště. Oboje má svůj smysl a svou definici.

4.1.1 Labyrint

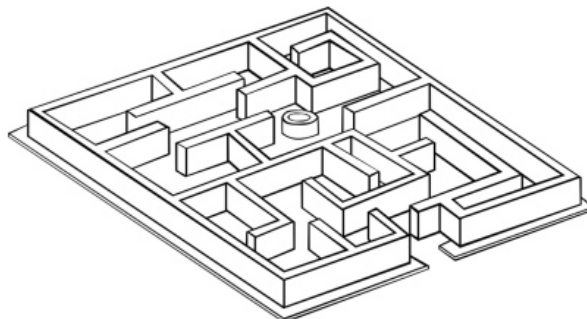
Labyrint (Obrázek 11) jako takový nemá žádnou možnost rozhodování. Cesta je jasně daná a vede k jasnému cíli. Většinou nebývá ani nějak výrazně oddělena např.: vysokými zdmi, ploty či jiným způsobem. Dokud účastník jde v před, vždy dojde na konec. [21]



Obrázek 11 Labyrint [21]

4.1.2 Bludiště

Bludiště se od labyrintu odlišuje svojí komplexností. Již se nejedná o jednosměrnou cestu k jasnému cíli, ale o spleť cest, slepých uliček, zatáček a stěn. V bludišti není vidět na jasný cíl hned od začátku, ale vždy cíl má a je řešitelný. Cesty jsou vymezeny vysokými stěnami, které mohou být tvořeny například křovím, plotem nebo třeba zrcadli. Účastník se zde může ztratit a zabloudit. O správnosti cesty rozhoduje náhoda a zkušenosti zde nehrají roli. [21]



Obrázek 12 Bludiště [21]

4.2 Definice bludiště

Pro následnou charakteristiku náročnosti bludiště je třeba si zavést určité prvky, ze kterých se bludiště skládá a podle kterých budeme určovat složitost bludiště.

Bludiště je cesta nebo soubor cest, který začíná vchodem do bludiště a končí v cíli, jako je střed bludiště nebo výstup z bludiště.

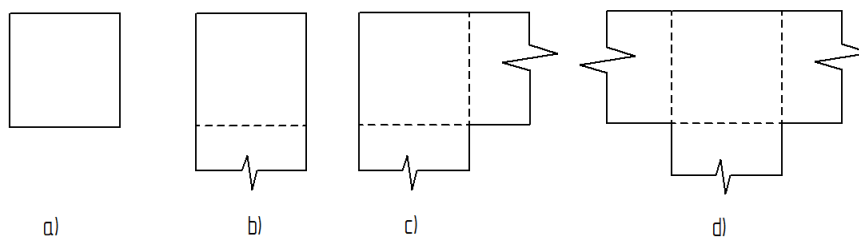
4.3 Základní prvky

Bludiště se skládá z několika základních prvků, jejichž kombinací vzniká samotné bludiště o dané náročnosti. [22]

- Buňka

Jedná se o nejmenší prvek bludiště. Obecně buňka nemá přesně definovaný tvar. Jak buňka vypadá, záleží na mozaice bludiště. Pro naši potřebu budeme využívat ortogonální mozaiku bludiště, tedy klasické čtvercové schéma.

V bludišti rozlišujeme čtyři typy buněk (Obrázek 13), izolovaná buňka (a), slepý konec (b), buňka jako součást cesty (c) a křižovatka (d).



Obrázek 13 Druhy buněk

- Chodba

Skládá se ze dvou zdí a cesty mezi nimi. Je tvořena souvislou posloupností buněk, které spojují křižovatku nebo slepý konec.

- Brána

V našem případě, je každé bludiště uzavřené a jako vstup či výstup bludiště slouží průhledné, barevně odlišené stěny, umístěné na jeho začátku a konci (Obrázek 51).

- Cesta

Posloupnost vzájemně propojených buněk v bludišti. Cest může být několik, nicméně my budeme cestu používat jako nejkratší možnou posloupnost buněk od začátku do konce bludiště.

- Smyčka

Cesta, která propojuje samu sebe a vytváří tak okruh.

- Slepý konec

Chodba, která má jeden konec zatarasený zdí. Pouze druhý konec je propojen se zbytkem bludiště.

4.4 Typy a řazení podle vzhledu

Bez znalosti struktury bludiště, ho stále můžeme rozdělit dle těchto vzhledových charakteristik [23]:

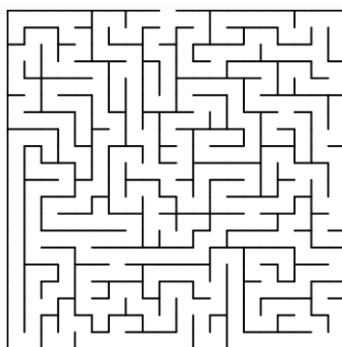
- Mozaika
- Dimenze
- Cesta

4.4.1 Rozdělení dle typu mozaiky

Mozaikou je myšlen samotný vzhled jednotlivých buněk. Bludiště se dá podle mozaiky rozdělit do několika druhů:

1) Ortoagonální

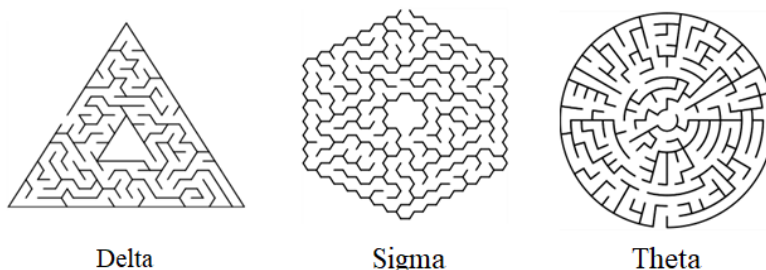
Klasické základní bludiště, jejichž síť se skládá z buněk, které mají cesty a křižovatky pod pravým úhlem. Také se těmto bludištím říká Gama bludiště (Obrázek 14).



Obrázek 14 Ortoagonální bludiště [24]

2) Omega

Pod pojem Omega bludiště spadají všechna neortoagonální bludiště. Základními podtypy Omega bludiště jsou Delta, Sigma a Theta bludiště (Obrázek 15). Nicméně sem se dá zařadit jakýkoli jiný typ mozaiky, který jsme schopni vymyslet.



Obrázek 15 Omega Bludiště [24]

3) Trhlinové

Tento typ bludišť nemá žádnou konzistentní síť. Jeho zdi a cesty jsou pod náhodnými úhly.

4) Fraktální

Mezi fraktální bludiště řadíme takové, u kterých se jednotlivé buňky stávají samotnými bludišti.

4.4.2 Rozdělení dle typu dimenze

Základními typy jsou 2D a 3D bludiště. Speciálními případy jsou pak více dimenzionální bludiště.

1) 2D

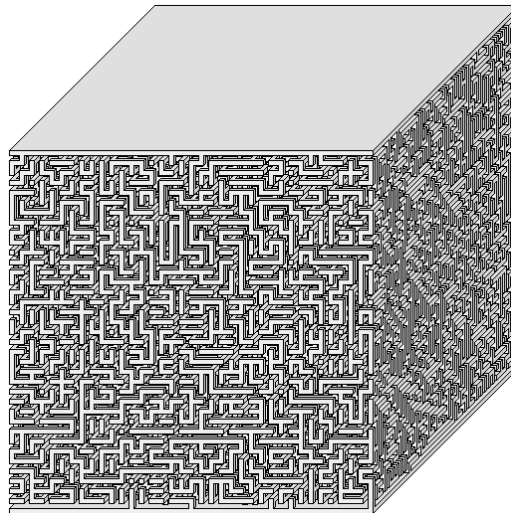
Většina bludišť je zobrazena v této podobě. Ať už na papíře nebo v realitě. V tomto případě je vždy možné jednoznačně nastínit cestu bludištěm, bez překrývání.

2) 3D

Toto bludiště má několik úrovní, které se navzájem mohou překrývat. Většinou jsou tyto bludiště zobrazena jako 2D síť s možnostmi jít nahoru či dolů.

3) Více dimenzionální bludiště

Z hlediska datových struktur není tvorba takových labyrintů problematická, ale jejich vizualizace je velmi složitá a pro běžného člověka poměrně těžko představitelná. Tento problém lze částečně vyřešit teleporty, které přepravují procházejícího jedince mezi bludišti (Obrázek 16).



Obrázek 16 Více dimenzionální bludiště [23]

4.4.3 Rozdělení dle typu cest

Toto rozdělení, je vzhledem k vytváření bludiště nejpodstatnější. Respektuje výše nadefinované geometrie bludiště.

1) Perfektní bludiště

Toto bludiště neobsahuje žádné smyčky nebo nepřístupná místa. Pravděpodobně se s ním člověk setká nejčastěji. Perfektní bludiště je nejčastěji používáno v dětských cvičebnicích. Také může být nazýváno jednoduše propojené bludiště. Z každého bodu bludiště je jen jedna cesta do jiného místa v bludišti. Má jen jedno řešení.

2) Zapletené bludiště

Neobsahuje žádné slepé konce. V tomto bludišti je využíváno zejména smyček, které se navzájem propojují, z toho také vyplývá jeho název. V závislosti na propojení jednotlivých smyček, bývá bludiště tohoto typu náročnější na vyřešení než bludiště perfektní, o stejné velikosti.

- 3) Jednocestné bludiště
Toto bludiště může být považováno za labyrint. Má jen jednu cestu a pokud se během procházení nezmění směr chůze, vždy řešitel dojde do konce.
- 4) Řídké bludiště
Zde existují buňky, ke kterým nejsou přivedené cesty a vytváří tak oddělené části v bludišti.
- 5) Částečně zapletené bludiště
Je zapletené bludiště, u kterého byla umožněna tvorba slepých konců.

4.4.4 Další rozdělení

Výše jsou uvedeny tři základní typy rozdělení. Nicméně u bludišť můžeme uvažovat i jiná rozdělení, jako například podle topologie⁹ daného bludiště. Například bludiště s normální topologií je bludiště vytvořené v klasickém Eukleidovském prostoru¹⁰. Na ostatní odlišné topologie referujeme jako na planární, tedy například topologie, kdy bludiště je vytvořeno na povrchu krychle. Další možností je pak textura, která charakterizuje bludiště bez ohledu na cesty či geometrii. Udává například gradient směru jednotlivých chodeb, jak dlouhé chodby jsou před tím, než jsou nějakým způsobem přerušeny nebo změny směr. Jak dlouhé je řešení vzhledem k velikosti bludiště a podobně. [23]

4.5 Shrnutí

Po představení jednotlivých možností v bludišti, se v této práci omezíme na perfektní ortogonální bludiště z důvodu lehčího rozdělení úrovní, úspory času a nevyvážené náročnosti ostatních druhů bludišť. Každé bludiště bude vytvořeno ve 3D.

⁹ Topologie bludiště – Týká se popisu geometrické struktury bludiště jako celku, a to bez ohledu na konkrétní uspořádání stěn, chodeb a cest uvnitř něj. Jedná se o abstraktní popis, který se soustředí na vztahy mezi různými oblastmi a průchody v bludišti, bez ohledu na jejich fyzickou podobu. Například topologie bludiště by mohla popsat, že existuje jedna cesta od vstupu do středu bludiště a zpět, ačkoli by neuváděla, jakým způsobem tato cesta vypadá nebo kudy vede.

¹⁰ Eukleidovský prostor – je metrickým prostorem, ve kterém zavádíme vzdálenost (každé dva body v prostoru mají mezi sebou určitou vzdálenost). Na tento prostor jsme intuitivně zvyklí.

5 Tvorba modelu bludiště

Pro tvorbu simulátorů se běžně využívá speciálních softwarů¹¹, které velmi usnadňují vývoj dané VR aplikace. Tyto softwary se od sebe liší například složitostí, funkcemi, ale také komunitou. Protože čím více lidí software používá, tím více lze nalézt různá řešení k daným problémům. Ostatní uživatelé pak ve spoustě případů sami vytváří šablony jejich řešení, které se následně mohou využít zbytkem komunity.

Pro realizaci řešení bylo zvoleno vývojové prostředí Unity3D.

5.1 Unity 3D

Tento nástroj je multiplatformní herní engine¹², vyvinutý společností Unity Technologies. Unity3D podporuje jak tvorbu ve 2D, tak i ve 3D. Jednou z hlavních předností tohoto engine je velká komunita vývojářů, díky které lze snadno s prostředím pracovat a dohledat spoustu informací ohledně funkčnosti a kódů pro naprogramování logiky celého VR, včetně propracovaných návodů k většině funkcí Unity. Tyto příspěvky komunity sjednocuje Asset Store¹³, přímo v prostředí Unity3D, kde lze stáhnout spoustu modelů k urychlení návrhu aplikace. Tohoto aspektu bylo využito i při tvorbě simulátoru bludiště využívaného pro testování. Engine je napsán v jazycích C, C++ a C# a umožňuje vyvíjet nové skripty v jazyce C#. Při dodržení licenčních podmínek je možné Unity3D využívat zcela zdarma. [25]

5.2 Generátor bludiště

Pro vytváření simulačních prostředí, tedy bludišť jednotlivých úrovní bude využit generátor zakoupený z Unity Asset Storu. Tento generátor je schopný vytvářet námi nadefinovaná bludiště viz kapitola Shrnutí.

5.2.1 Porovnání generátorů bludiště

Při výběru vhodného nástroje pro tvorbu bludišť jsme využili Unity Asset Store, dále jen UAS, kde byli vyhledány generátory bludišť pod názvem "maze generator". Výsledkem hledání bylo 36 výsledků od komunitních tvůrců, z nichž čtyři byly zdarma. Po pečlivém prostudování nabídky byl vybrán nástroj, který nejlépe vyhovoval požadavkům práce.

Hlavním kritériem při výběru vhodného generátoru bludišť z UAS bylo splnění určených požadavků. Byla posouzena cena, uživatelská přívětivost, přizpůsobitelnost, možnost přizpůsobení vzhledu, podrobnost návodu a nadstavbová funkce "path-finding"¹⁴. Z nabízených 36 možností byly vybrány tři nejvhodnější:

- 1) Umbra Maze Magician od Umbra Evolution [26]
- 2) Core Maze od BatSoft [27]
- 3) Custom Maze Generator od Jesterway Software. [28]

Tyto tři generátory bludišť nejlépe splňovaly požadavky a byly dále podrobně porovnány.

¹¹ Software nebo také programové vybavení, je nehmotná část počítače, která provádí nějakou činnost.

¹² Jako engine se v herním vývoji označuje program, ve kterém se vytváří daná hra. Jde o komplexní a víceméně jednotný soubor nástrojů pro vývojáře, pomocí kterých hra získává jednotný tvar a může fungovat jako celek.

¹³ Online knihovna doplňků k vývojovému prostředí Unity.

¹⁴ Path-finding – Jedná se o nadstavbovou funkci, kterou mohou mít některé generátory bludišť. Je to algoritmus, který určuje, jaká cesta vede z počátečního bodu do cíle. Tato funkce umožňuje uživateli lépe definovat složitost bludiště tím, že určí nejkratší cestu mezi počátečním a cílovým bodem.

Hodnocení je reprezentováno body, kdy nejhodnější generátor pak získá největší průměrné hodnocení.

Tabulka 1 Hodnocení generátorů

Kritérium	Umbra Maze Magician	Core Maze	Custom Maze Generator
Cena	25 USD	50 USD	15 USD
Uživatelská přívětivost	5/5	4/5	3/5
Přizpůsobitelnost	4/5	4/5	3/5
Nastavení vzhledu	5/5	4/5	3/5
Path-finding	ANO	ANO	NE
Celkové hodnocení	4,5/5	3,7/5	3/5
Pořadí	1.	2.	3.

Z Tabulka 1 vyplývá, že Umbra Maze Magician od Umbra Evolution je nejlepší volbou, díky jeho nejlepší uživatelské přívětivosti, přizpůsobitelnosti a nastavení vzhledu. Navíc obsahuje funkci path-finding. Je dražší než některé z ostatních generátorů, ale jeho kvalita a funkce jsou výborné.

5.2.2 Umbra Maze Magician

Tento generátor nabízí uživatelsky přívětivé rozhraní, které umožňuje snadnou práci při nastavování parametrů bludiště, jako jsou jeho rozměry, velikost, typ cest, vzhled jednotlivých stěn a opakované generování téhož bludiště. Dále poskytuje detailní a srozumitelný videonávod k použití.

Autor Umbra Maze Magician dále pravidelně aktualizuje a vylepšuje tento generátor, což je důležité s ohledem na možné budoucí úpravy a rozšíření testování. Autor je také velmi aktivní a rychle reaguje na dotazy a problémy uživatelů, což přineslo větší jistotu při použití tohoto generátoru.

5.2.3 Seznámení s UMM

UMM generátor je možné přidat do scény dvěma způsoby. Prvním je vložení UMM editoru přes okno „Project“, kde po rozvětvení složky „UmbraEvolution“ se dostaneme, až k příkladové scéně, kterou již vytvořil autor generátoru. Tato scéna má v sobě i další pro nás nepotřebné objekty, jako je například již vytvořené bludiště, kamera, světla apod.

Zvolili jsme tedy druhý způsob přidání UMM generátoru, a to je přes horní záložku „Tools“, která se vytvořila po importu UMM assetu.

„Tools“ → „UmbraEvolution“ → „Umbra Maze Magician“ → „Create Maze Generator“

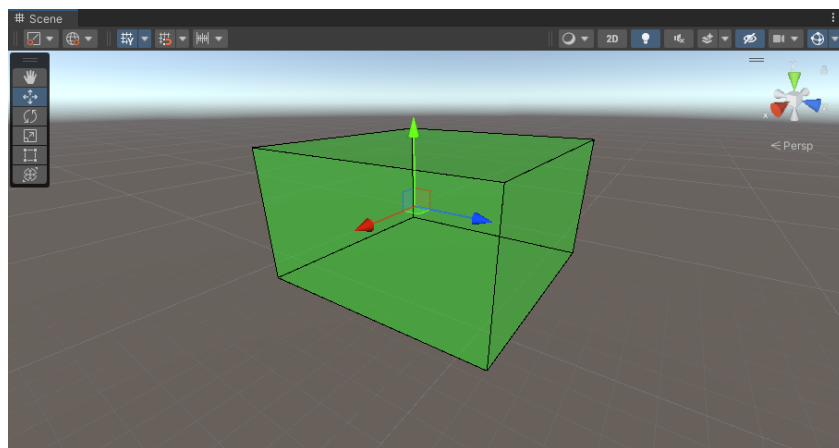


Obrázek 17 Create Maze Generator

Tímto krokem se do scény přidal GameObject „UmbraMazeMagician“, který slouží jako editor pro tvorbu bludišť.

Editor obsahuje následující nastavení:

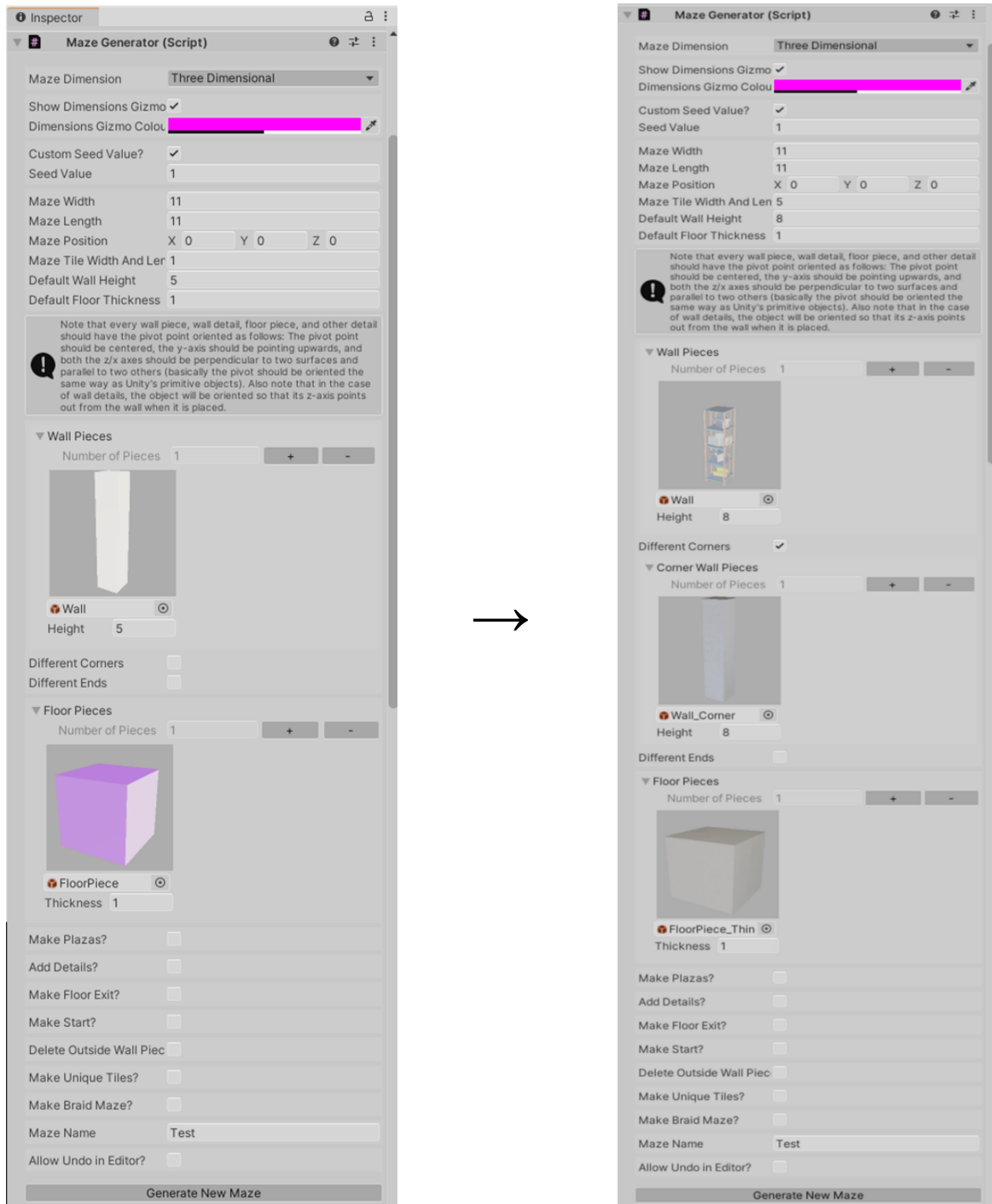
- Maze Dimension* – Určuje, zda se vytvoří 2D nebo 3D bludiště.
- Show Dimension Gizmo* – Naznačí ve Scene View obrys bludiště (Obrázek 18).
- Dimensions Gizmo Colour* – Nastavení barvy obrysu bludiště.



Obrázek 18 Gizmo

- Custom Seed Value* – Umožňuje vygenerovat bludiště na základě předem nastaveného seedu. Z bludiště, které má vyhovující vlastnosti, si stačí tedy uložit pouze hodnotu seedu a jeho rozměry. Generátor je pak schopný podle těchto údajů vygenerovat stejné bludiště. Tato vlastnost bude dále využita.
- Maze Width* – Určuje celkovou šířku generovaného bludiště.
- Maze Length* – Určuje celkovou délku generovaného bludiště.
- Maze position* – Určuje pozici generovaného bludiště ve scéně.
- Maze Tile Width And Length* – Určuje šířku a délku jednotlivých polí, ze kterých se bludiště skládá.
- Default Wall Height* – Nastavuje výšku Gizma.
- Default Floor Thinckness* – Nastavuje, v jaké výšce se bude generovat podlaha bludiště.
- Wall piece* – Do tohoto pole přidáváme prefab stěn, ze kterých bude bludiště tvořeno. Pokud by bylo využito více stěn, pomocí plusového tlačítka, se může přidat více možných prefabů, ze kterých se pak náhodně stěny vytvoří.
- Different Corners* – Umožňuje přidat no rohy křížovatek v bludišti jiné stěny. V našem případě to bude model betonového sloupu.
- Different Ends* – Umožňuje nastavit jiný prefab pro zakončení slepých uliček.
- Floor Pieces* – Zde se nastaví prefab pro vzhled podlahy. Opět je možné nastavit více prefabů, ze kterých se pak náhodně vygenerují jednotlivá políčka podlahy.
- Make Plazas?* – Umožňuje generovat v bludišti prostor bez stěn. Tento prvek může sloužit k vytvoření prostoru s více možnostmi rozhodnutí. Pro naše bludiště jsme této možnosti nevyužili.
- Add Details?* – Nastavení, které umožňuje rozmístit po bludišti například lampy na stěny apod.

- Make Floor Exit?* – Přidává východ z bludiště v podobě otvoru v zemi.
- Make Start?* – Nadefinuje zvlášť prefab pro startovní pole bludiště.
- Delete Outside Wall Piece* – Umožňuje vytvořit mezery mezi stěnami ohraňujícími bludiště.
- Make Unique Tiles* – Přidává do bludiště specializovaná políčka.
- Make Braid Maze* – Pokud je toto pole zaškrtnuto, odstraní se z bludiště slepé konce.
- Maze Name* – Jméno, které bude přiřazeno GameObjectu pod, kterým bude prefab bludiště umístěn.
- Allow Undo in Editor?* – Dává Unity přístup využívat funkci „Undo“ v prostředí UMM editoru.



Obrázek 19 Nastavení v UMM

6 Klasifikace bludiště

Klasifikace bludišť je složitý proces, který zahrnuje různé přístupy. Vyhodnocení náročnosti bludiště je poměrně složitý matematický problém, který se dá řešit například pomocí teorie grafů. Existují i jednodušší způsoby, jak určit náročnost bludiště, například na základě počtu kroků, které jsou třeba podstoupit od začátku do konce, dle jeho velikosti a dalšími způsoby. Tyto způsoby však nemusí zahrnovat všechny faktory, které ovlivňují náročnost bludiště a nemusí tak být nejpřesnější.

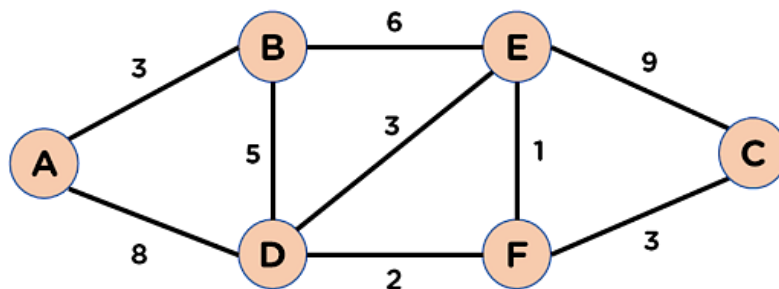
6.1 Klasifikace dle teorie grafů

Teorie grafů je užitečným nástrojem pro studium složitosti bludišť, protože umožňuje převést strukturu bludiště na abstraktní graf s uzly a hranami. Tento graf lze pak použít pro analýzu různých vlastností bludiště, jako je počet cest, délka nejkratší cesty, počet křižovatek nebo minimální počet stěn potřebných k oddělení bludiště na samostatné oblasti. K řešení těchto problémů lze využít různé algoritmy z oblasti teorie grafů, například Dijkstrův algoritmus nebo algoritmus A*, které umožňují najít nejkratší cestu v grafu zadaném bludištěm. Použití teorie grafů k řešení složitosti bludišť tak umožňuje efektivnější analýzu a výpočet různých vlastností bludišť, což má široké využití v oblastech jako jsou robotika, počítačové hry nebo navigační systémy.

6.1.1 Dijkstrův algoritmus

Tento algoritmus hledá nejkratší cestu mezi dvěma body v bludišti a dává tak návod na to, jak se z jednoho bodu dostat do druhého s minimálním počtem kroků. V případě bludiště se může tato cesta použít k nalezení cesty z počátečního bodu (např. vstupu do bludiště) do cílového bodu (např. východu z bludiště).

Dijkstraův algoritmus (Obrázek 20) hledá nejkratší cestu ze všech dosud nezpracovaných uzlů do nejbližšího uzlu, přičemž vzdálenost od počátečního uzlu se postupně zvyšuje. Pokud se cílový uzel nachází v dosahu, algoritmus končí a vrací nejkratší cestu. V případě bludiště by se uzly mohly reprezentovat jednotlivými buňkami bludiště. [29]



Obrázek 20 Dijkstrův algoritmus

6.1.2 A* a B*

Tyto algoritmy jsou modifikacemi Dijkstraova algoritmu, které používají heuristické funkce¹⁵ k rychlejšímu nalezení nejkratší cesty. Tyto funkce odhadují vzdálenost od aktuálního uzlu k cílovému uzlu. Algoritmus se pak snaží hledat cestu, která má nejnížší celkovou odhadovanou vzdálenost od počátečního uzlu k cílovému uzlu. Tyto algoritmy se obvykle používají v

¹⁵ Heuristická funkce – Je funkce používaná v některých algoritmech, která slouží k odhadnutí toho, jak blízko jsme k řešení problému.

aplikacích, které vyžadují nalezení nejkratší cesty v reálném čase, např. navigační aplikace nebo videohry.

V případě bludiště by se mohly A* a B* algoritmy použít tak, že by se každá buňka bludiště zobrazil na grafu a hledání nejkratší cesty by začalo na počáteční buňce a pokračovalo by přes sousední buňky, dokud není dosažen cílový uzel. Výsledkem by byla nejkratší cesta z počátečního bodu do cílového bodu v bludišti. [30]

6.1.3 Další algoritmy

Další teorie grafů, které lze využít pro určení složitosti bludišť, zahrnují například algoritmy pro detekci cyklů a hledání minimální kostry grafu nebo teorie perkolace, která se zaměřuje na pravděpodobnost toho, že se daný bod v bludišti spojí s jiným bodem. Pomocí této teorie lze vypočítat, jaká je pravděpodobnost, že se například v bludišti s určitou hustotou překážek podaří najít cestu mezi dvěma body. Tyto algoritmy se obvykle používají k nalezení nejkratší cesty v grafu a mohou být aplikovány i na bludiště. [31]

6.2 Klasifikace na základě geometrie

Tato kategorie klasifikace se zaměřuje na jednodušší metody určování náročnosti bludiště. Tyto metody se soustředí přímo na analýzu vzhledu bludiště a vycházejí ze subjektivního rozdělení úrovně bludiště na základě jeho velikosti, počtu řešení, počtu možných rozhodnutí při průchodu bludištěm, délky cesty řešení v určených jednotkách a podobně.

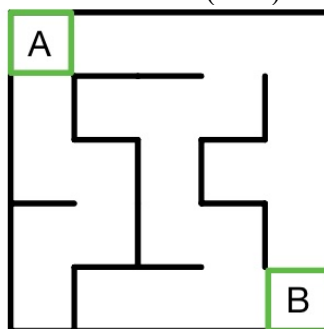
Složitost tedy můžeme určit například na základě počtu kroků, které jsou třeba podstoupit od začátku do konce. Tento postup je intuitivní, ale nemusí zahrnovat všechny faktory, které ovlivňují náročnost bludiště. Nicméně, tento způsob měření náročnosti bludiště je poměrně jednoduchý a často se používá jako základní metoda pro porovnávání různých typů bludišť. Další jednodušší postupy pro určení náročnosti bludiště zahrnují měření počtu rozcestí a slepých uliček nebo velikost nejdelší cesty v bludišti. Tyto postupy mohou být využity k určení náročnosti bludiště s menší přesností než metoda měření počtu kroků, ale jsou o něco jednodušší k provedení a poskytují základní představu o náročnosti bludiště.

Postupy

1) Počet kroků

Jedním ze způsobů, jak určit složitost bludiště, je na základě počtu kroků vedoucích k jeho řešení. Tento postup spočívá v tom, že pro každé bludiště se spočítá počet kroků (tj. posunutí ze startu do cíle), které jsou nutné k jeho úspěšnému vyřešení.

Příkladem může být bludiště o velikosti 5x5 (Obrázek 21), které začíná v levém horním rohu a končí v pravém dolním rohu. V následujícím bludišti jsou některé cesty blokovány a hráč se musí dostat z bodu A (start) do bodu B (cíl):



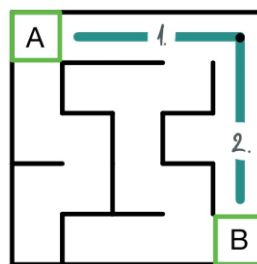
Obrázek 21 Bludiště 5x5

Pro řešení tohoto bludiště existuje několik možností, ale pro jednoduchost uvažme cestu vpravo a dolů. Tato cesta je jedna z nejkratších, které vedou z bodu A do bodu B, a sestává ze dvou kroků (pohyb vpravo se počítá jako jeden krok a pohyb dolů se také počítá jako jeden krok).

Podobně se spočítá počet kroků i pro ostatní cesty vedoucí z bodu A do bodu B. Vyhodnocení složitosti bludiště pak spočívá v tom, že se porovnávají tyto počty kroků pro různá bludiště a hledá se nejkratší cesta. Počet kroků nejkratší cesty pak vytvoří stupnici, dle které můžeme subjektivně rozřadit jednotlivé bludiště dle jejich náročnosti.

2) Počet rovných úseků

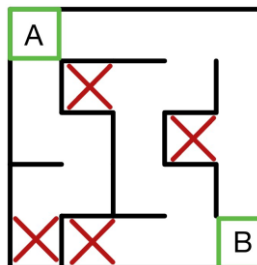
Bludiště lze rozdělit na rovné úseky, což znamená, že úsek končí, když se cesta bludiště otočí o 90 stupňů. Náročnější bludiště bude mít více takových úseků, zatímco jednodušší bludiště budou mít méně úseků. Pro naše bludiště o velikosti 5x5, pak vychází dva úseky.



Obrázek 22 Bludiště 5x5 – úseky

3) Počet slepých konců

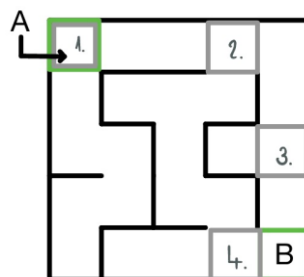
Slepé konce jsou body v bludišti, kde cesta končí bez možnosti pokračovat dál. Jednodušší bludiště budou mít méně slepých konců, zatímco náročnější bludiště budou mít více slepých konců.



Obrázek 23 Bludiště 5x5 – slepé konce

4) Počet rozcestníků

Rozcestníky jsou body v bludišti, kde se cesta větví na více možností. Jednodušší bludiště budou mít méně rozcestníků, zatímco náročnější bludiště budou mít více rozcestníků.



Obrázek 24 Bludiště 5x5 – rozcestníky

Tyto způsoby mohou být užitečné pro rychlou aproximaci náročnosti bludiště a mohou být použity v situacích, kdy není potřeba přesné řešení. Pokud bychom potřebovali přesnější výsledky, může být lepší použít složitější metody, jako je teorie grafů.

Nicméně, pro účely této práce, je využití teorie grafů příliš složité a zbytečné. V tomto případě je vhodnější využít jednodušší a praktičtější postup hodnocení dle geometrie.

6.3 Stanovení metody klasifikace

Pro testování, je zapotřebí vytvořit bludiště pro každý druh ovládání. Pro jednotlivá ovládání jsou potřeba 3 samostatné bludiště o úrovních I, II, III, které jdou od nejjednodušší po nejtěžší. Je tedy zapotřebí vytvořit celkem 9 bludišť, jakožto simulačních prostředí.

6.4 Parametry

Vytvoříme tak subjektivní hodnocení, na základně rozhodujících parametrů.

Jako rozhodující parametry budeme uvažovat tyto:

1) Velikost bludiště

Bludiště budou vždy čtvercová. Pro účely testování není zapotřebí vytvářet zvláště velká bludiště. Uvažujeme-li velikost jedné buňky jako 1x1, pak velikost bludiště kategorie I bude 11x11. Pro kategorie II a III se pak rozměr zvětší o krok 4 viz Tabulka 2.

Tabulka 2 Rozdělení dle velikosti

Kategorie	I	II	III
Velikost	11x11	15x15	19x19

2) Počet kroků

Počet kroků potřebný k dosažení cíle. Tento parametr by měl přímo úměrně narůstat s velikostí bludiště a bude zejména zajišťovat shodnou náročnost mezi bludišti ve stejných kategoriích. Pro jednotlivé kategorie je pak určen počet kroků, který by se měl u všech bludišť, napříč ovládaními, shodovat.

3) Počet slepých konců

Počet slepých konců slouží opět, jako kontrolní ukazatel stejné náročnosti mezi bludišti pro odlišné ovládaní ve stejné kategorii. Tento parametr vyznačuje možný počet chyb, které může uživatel při průchodu prostředím udělat.

4) Délka cesty

Délka cesty je dalším parametrem, který přispívá k určení náročnosti bludiště. Tento parametr vyjadřuje celkovou délku nejkratší cesty, kterou musí uživatel projít, aby dosáhl cíle. Během této cesty by uživatel měl být schopen obejít všechny slepé konce a vyhnout se zbytečnému přecházení stejných míst. Stejně jako u počtu kroků a slepých konců, i délka cesty by měla být stejná, pro bludiště ve stejných kategoriích.

Konkrétní parametry pro jednotlivé kategorie jsou pak definovány v Tabulka 6.

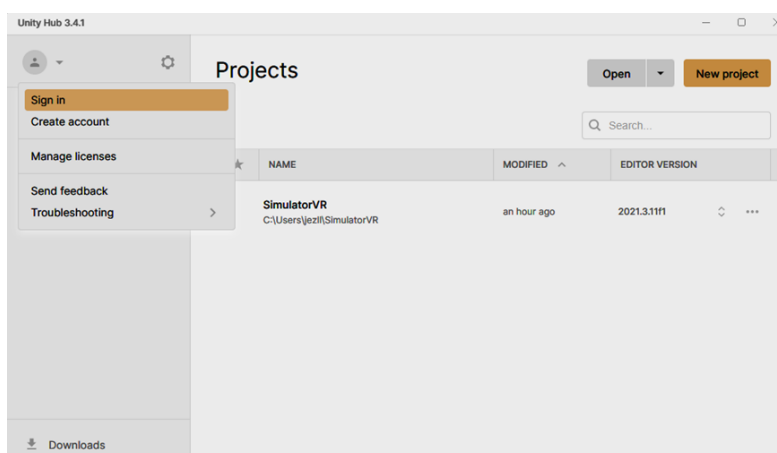
7 Tvorba simulátorů ve virtuální realitě

Jedním z cílů této práce bylo vytvořit simulační prostředí pro VR, ve kterém bude možné jednotlivá ovládaní otestovat. Jak bylo výše zmíněno v kapitole 5.2.2, k tomu účelu napomůže vytvořený generátor bludišť, který funguje v softwaru Unity3D.

Bylo nutné tedy samotné Unity stáhnout a nainstalovat, k tomu jsme využily oficiální web [Unity](#).

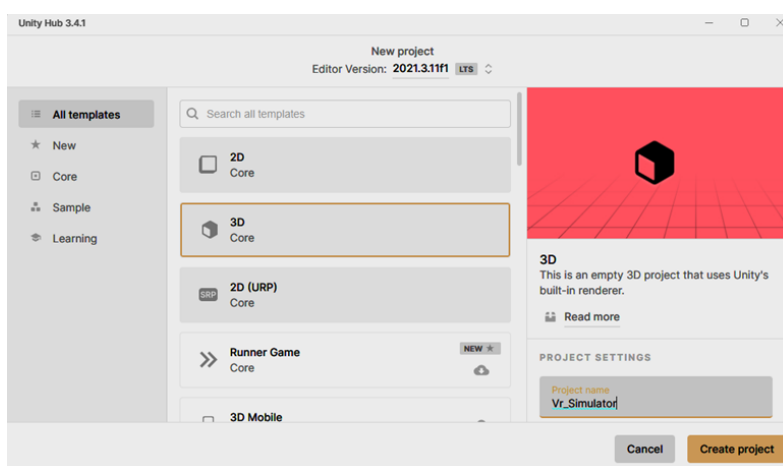
7.1 Vytvoření projektu

Zároveň je nedílnou součástí připojit k Unity účet, který byl dříve založen na webu [Unity Asset Store](#). Díky tomuto propojení, se synchronizují zakoupené assety¹⁶, které využijeme v projektu. Toto přihlášení provedeme přímo v aplikaci Unity Hub viz Obrázek 25.



Obrázek 25 Přihlášení

V aplikaci Unity Hub byl založen nový projekt prostřednictvím tlačítka „New Project“, který využívá 3D template¹⁷ a pojmenován „Vr_Simulator“.



Obrázek 26 Zakládání projektu

¹⁶ Asset – V kontextu Unity, Asset znamená jakýkoliv soubor, který se používá v projektu a přidává se do něj. To může být například 3D model, textura, zvukový soubor, skript nebo animace.

¹⁷ Template – V Unity projektu se Template (šablona) obvykle odkazuje na předdefinovaný soubor projektu, který obsahuje několik základních prvků a nastavení, které se používají při vytváření nového projektu.

7.1.1 Seznámení s uživatelským prostředím unity

Po vytvoření nového projektu se dostaneme do uživatelského rozhraní Unity Editoru viz Obrázek 27.

V tomto prostředí se upravuje a postupně buduje celý projekt. Mezi jeho hlavní prvky, které se využívají k práci patří:

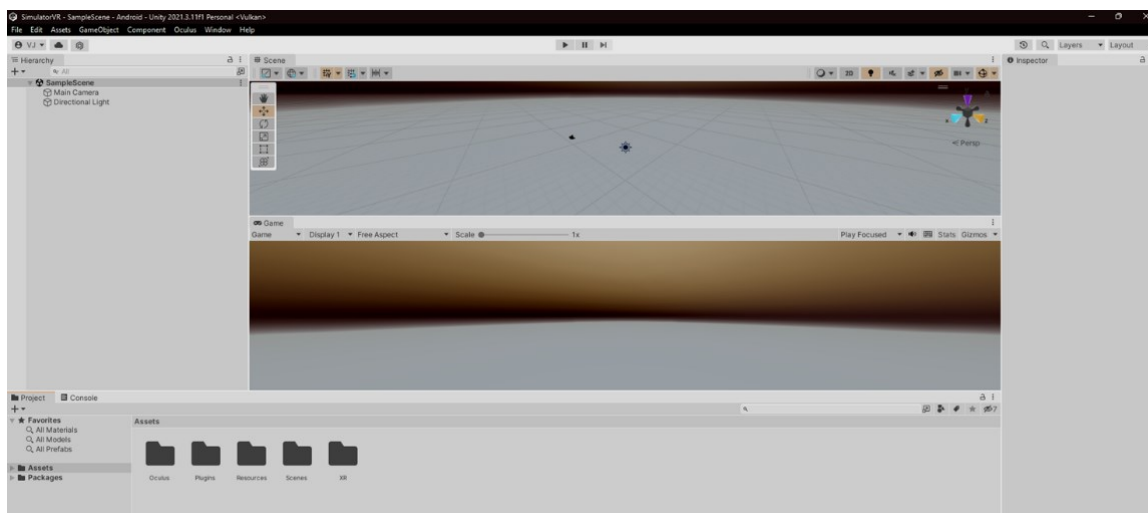
- 1) Hierarchy
Slouží k zobrazení hierarchie objektů v aktuální scéně. Jedná se o panel v editoru, který ukazuje seznam všech objektů, které jsou momentálně umístěny v aktuální scéně¹⁸. Tento panel zobrazuje strukturu objektů jako strom, kde rodičovské objekty jsou na vrcholu a potomci jsou pod nimi. Zároveň slouží jako nástroj pro manipulaci s objekty v aktuální scéně. Umožňuje vývojářům přidávat, odebírat, přesouvat, organizovat nebo vyhledávat objekty a komponenty v rámci scény.
- 2) Scene view
Je jedním z pohledů v Unity Editoru, který zobrazuje aktuálně vybranou scénu ve 3D prostoru. V Scene view mohou vývojáři přidávat, odebírat a upravovat objekty, prvky, osvětlení a zvuky v dané scéně. Můžou také měnit perspektivu a polohu kamery, aby mohli pracovat s jednotlivými prvky v prostoru.
- 3) Game view
Zobrazuje aktuální scénu tak, jak ji uvidí hráči nebo uživatelé aplikace.
- 4) Inspector
V Inspectoru lze upravovat vlastnosti objektů a komponent, jako jsou pozice, rotace, měřítko, materiály, osvětlení, zvuky a další prvky. Vývojáři mohou také přidávat a odebírat komponenty, připojovat skripty¹⁹ a upravovat různé nastavení projektu.
- 5) Project
Obsahuje všechny assety a soubory, které jsou používány ve hře. V Project jsou uloženy zdrojové soubory, jako jsou skripty, modely, textury, zvuky a další prvky, které jsou součástí herního projektu. Vývojáři mohou přidávat nové assety, mazat je nebo je upravovat, přesouvat je do složek a organizovat je podle potřeby.
- 6) Console
Zobrazuje různé informace o běhu hry a o chybách, které se vyskytly během hraní. Console umožňuje sledovat chyby a varování v reálném čase. Konkrétně umožňuje zobrazovat informace o běhu skriptů, zprávy z pluginů²⁰, výstup z Debug.Log²¹, varování a chyby v Unity Editoru.
V práci consoli využijeme pro výpis délky cesty viz Kapitola 6.4.

¹⁸ Scéna – Unity je to prostředí, ve kterém se odehrává jedna herní úroveň. Mohou se zde vytvářet různé scény a přepínat mezi nimi během hraní hry. Každá scéna může mít svůj vlastní pohled kamery a své vlastní nastavení.

¹⁹ Skript – V Unity, je skript programovací kód napsaný v jazyce C#, JavaScript nebo Boo, který umožňuje interakci a manipulaci s prvky ve scéně hry. Skripty mohou být připojeny k různým objektům ve scéně, jako jsou herní objekty, kamery a další prvky, a umožňují programovat chování a akce pro tyto objekty.

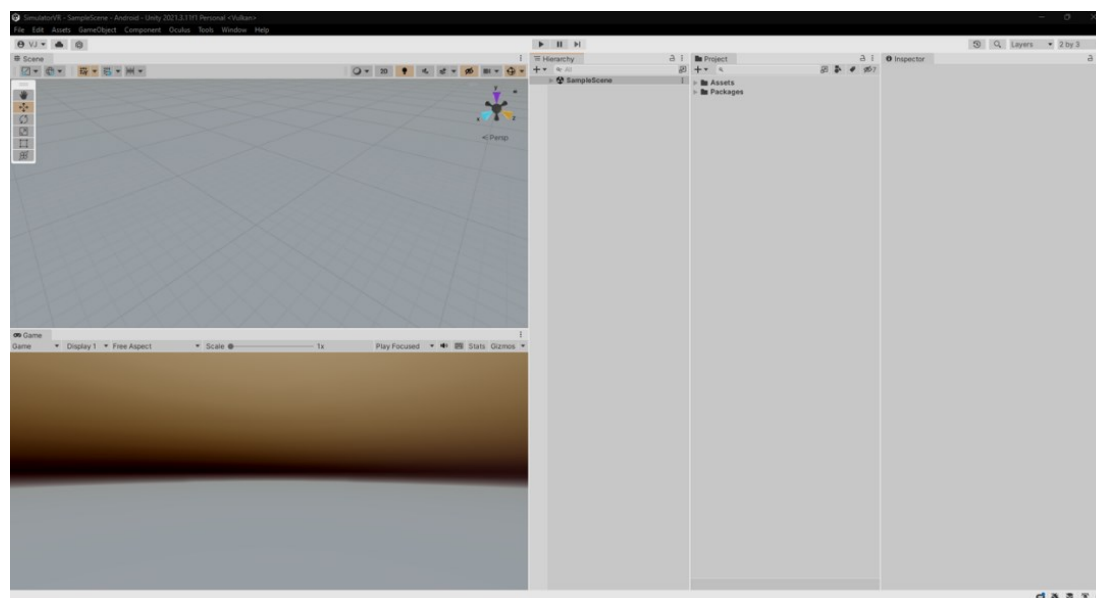
²⁰ Plugin – Je softwarový modul, který rozšiřuje funkčnost určitého programu. V kontextu Unity, plugin je soubor nebo sada souborů, které jsou přidány do projektu a slouží k rozšíření funkčnosti editoru nebo aplikace. Tyto soubory mohou obsahovat předdefinované funkce, komponenty a nástroje, které mohou být použity k vytváření nových funkcí a komponent pro hru. V Unity se často zaměňují za assety.

²¹ Debug.Log – Je funkce v Unity API, která umožňuje vkládat různé zprávy a informace do Console v Unity Editoru během vývoje hry.



Obrázek 27 Unity Editor

Pro snadnější práci a manipulaci s projektem a jeho větší přehlednost, změním rozložení pracovního prostoru editoru. V pravém horním rohu v záložce „Layout“ nastavíme „2 by 3“ a změním sloupec „Project“ na jednosloupcový.



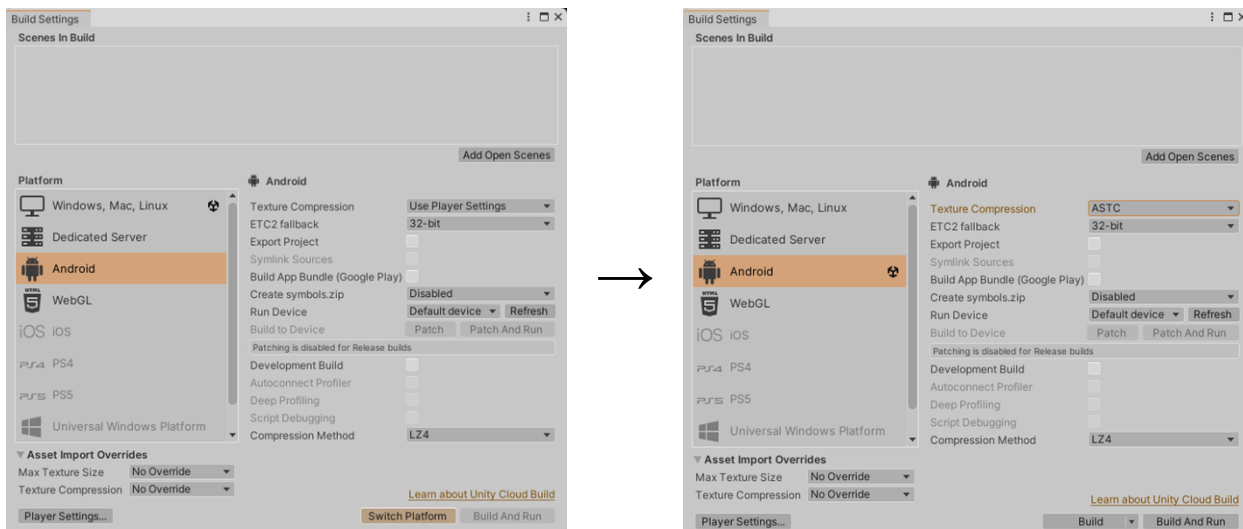
Obrázek 28 Unity Editor – přizpůsobený

7.1.2 Nastavení projektu

V první řadě je důležité správně nastavit celý Unity projekt hned zpočátku budování aplikace z důvodu správné optimalizace jednotlivých assetů, tak aby vhodně pracovala s VR brýlemi. Vyhne se tak případným problémům při finálním exportu aplikace do brýlí, kdy by Unity muselo provádět veškeré změny na již vytvořených prvcích, což by mohlo vést k nežádoucím errorům a velké časové náročnosti.

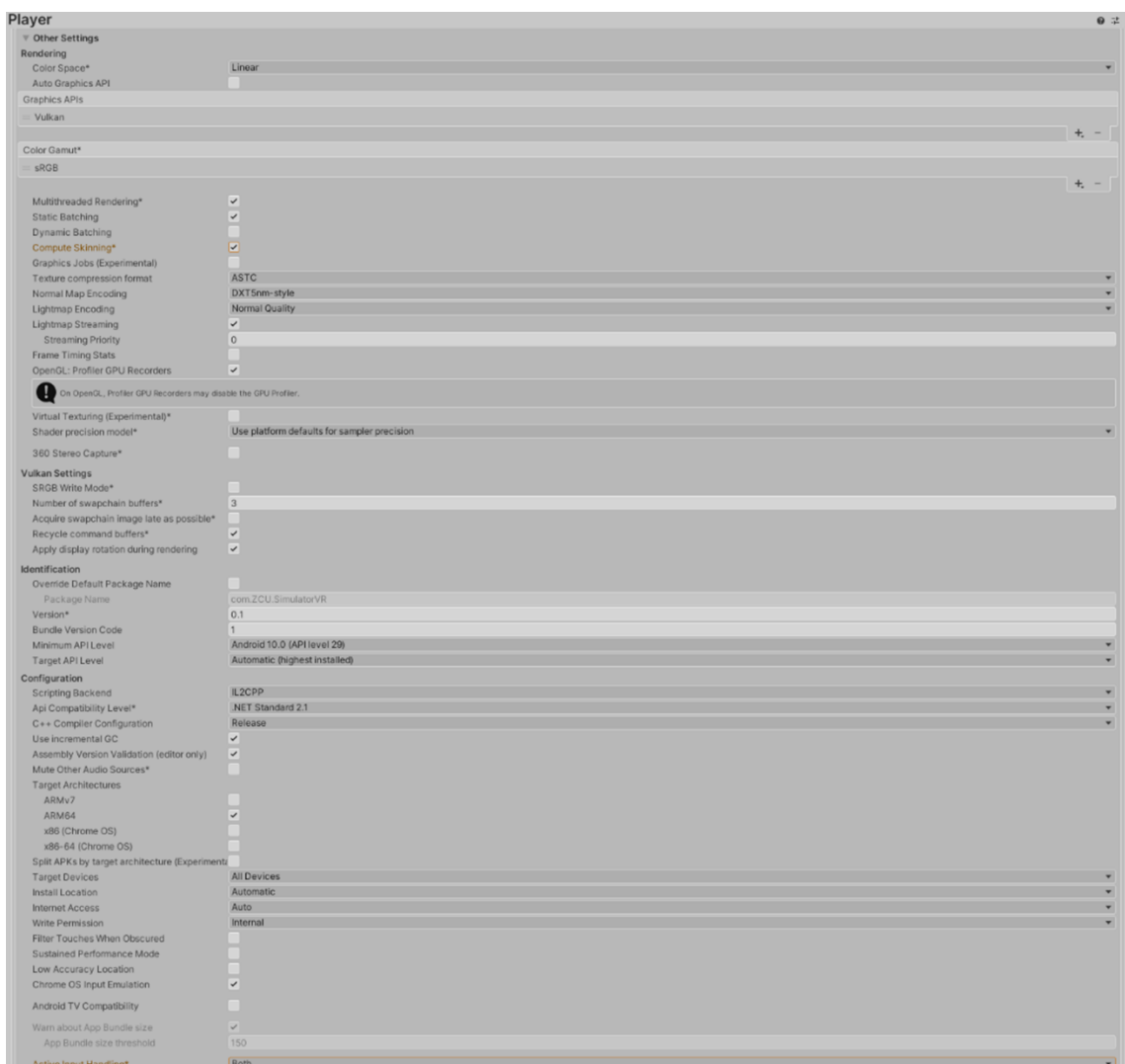
Jako první krok přepneme tedy platformu projektu. Tlačítko v horním menu „File“ → „Build Settings“ a z „Windows, Mac, Linux“ přepneme na platformu „Android“ a nastavíme „Texture Compression“ na „ASTC“²².

²² ASTC (Adaptive Scalable Texture Compression) – je moderní kompresní formát pro textury, který byl vytvořen s cílem snížit nároky na paměť, spotřebu baterie a zvýšit výkon na mobilních zařízeních.



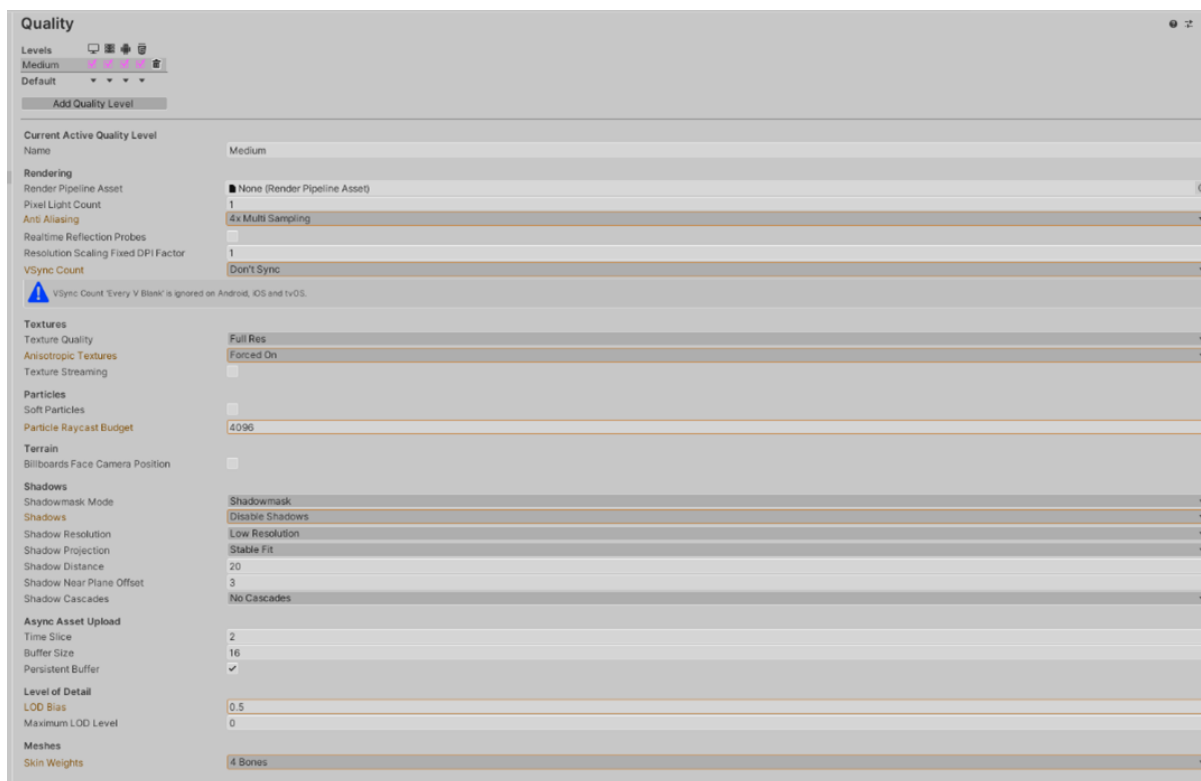
Obrázek 29 Build Settings

Jako další krok je důležité nastavit prvky v záložce „Project Settings“. V horní liště rozklikneme „Edit“ → „Project Settings“ → „Player“ → „Other Settings“, zde zaškrtneme políčko Compute Skinning* a změníme hodnotu Active Input Handling* na „Both“.



Obrázek 30 Project Settings – Player

V záložce „Quality“ změním „Levels“ na „Medium“, „Anti Aliasing“ na „4x Multi Sampling“, „VSync Count“ na „Don't Sync“, „Anisotropic Textures“ na „Forced On“, „Particle Raycast Budget“ na hodnotu 4096, „Shadows“ na „Disable Shadows“, „Lod Bias“ na hodnotu 0,5 a „Skin Weights“ na „4 Bones“.



Obrázek 31 Project Settings – Quality

Jako poslední krok zaškrtneme v „XR Plug-in Management“ políčko „Oculus“.



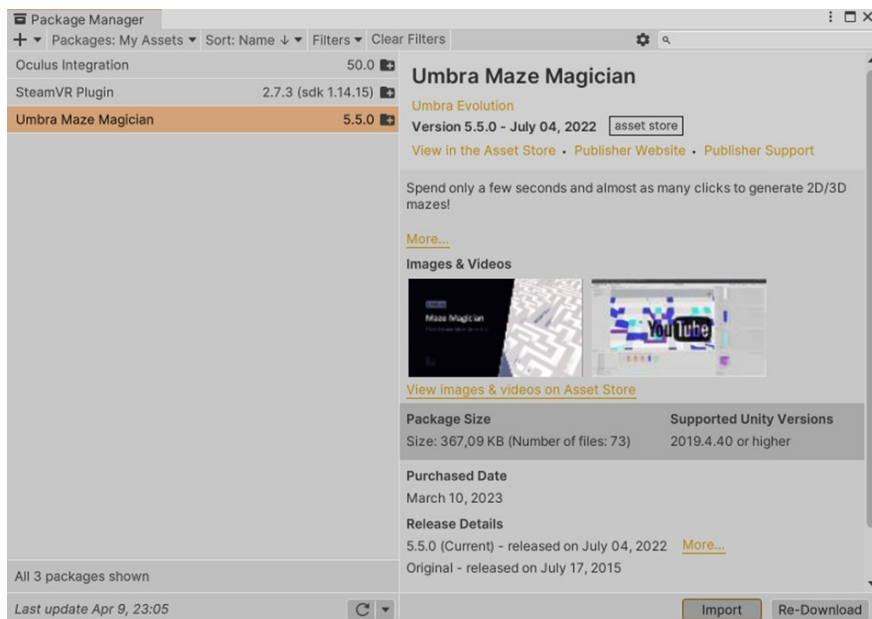
Obrázek 32 Project Setting – XR Plug-in

7.1.3 Instalace Assetu

Po seznámení s celkovým Editorem Unity a nastavení projektu, přišla na řadu samotná instalace veškerých Assetů (pluginů chceme-li) potřebných k vytvoření a zprovoznění simulátoru a správného provázání s VR headsetem.

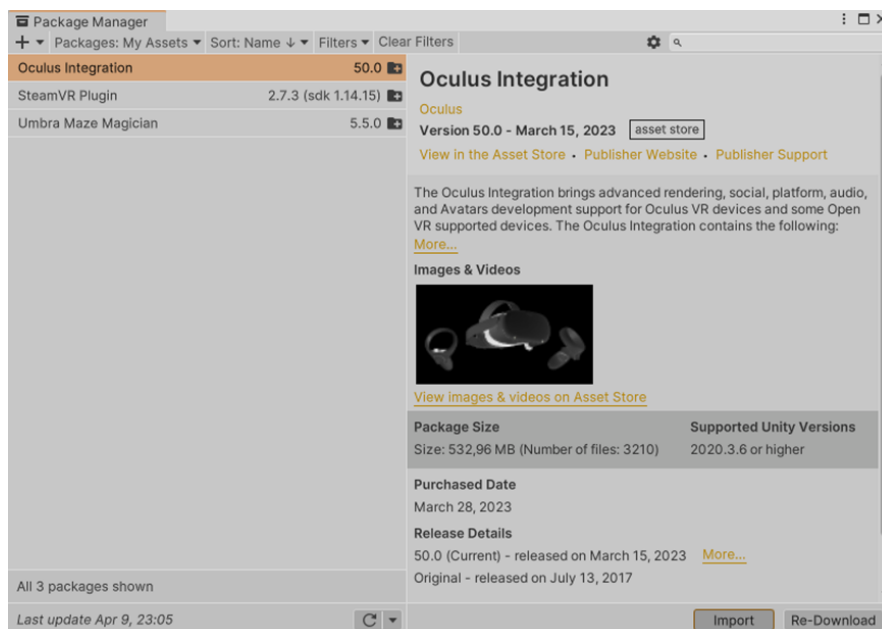
V první řadě bylo potřeba nainstalovat samotný generátor bludišť, který jsme si dříve zakoupili na unity asset store. První krok se odehrává ještě na webu unity, kde dostaneme možnost si po zakoupení určitého assetu, přidat tento asset mezi své vlastní, které se ukládají do našeho unity účtu.

Poté se tedy přesuneme do unity editoru, kde se musíme dostat do Package Mangeru. Tedy v horním menu tlačítko „Window“ → „Package manager“. V Package Manageru si v horním poli zaškrtneme políčko „My Assets“ a nainstalujeme asset Umbra Maze Magician.



Obrázek 33 Package Manager – UMM

Stejný postup aplikujeme pro Asset od společnosti Meta, Oculus Integration. Tento Asset je klíčovým nástrojem pro vývoj VR aplikací na platformě Oculus, protože umožňuje integraci všech potřebných funkcí a vlastností, které jsou nezbytné pro plnohodnotný VR zážitek. Součástí jsou již vytvořené a nascriptované modely rukou, ovladačů, tzv. prefab²³, kamer apod., které v mém simulátoru využiji. Jako poslední pak přidáme „SteamVR plugin“, který umožní komunikaci mezi aplikací a Cybershoes.



Obrázek 34 Package Manager – OI

²³ Prefab – Unity je to předdefinovaný objekt, který obsahuje určité komponenty, herní objekty a nastavení. Lze si ho představit jako šablonu, která se používá pro opakované vytváření stejných objektů v rámci hry.

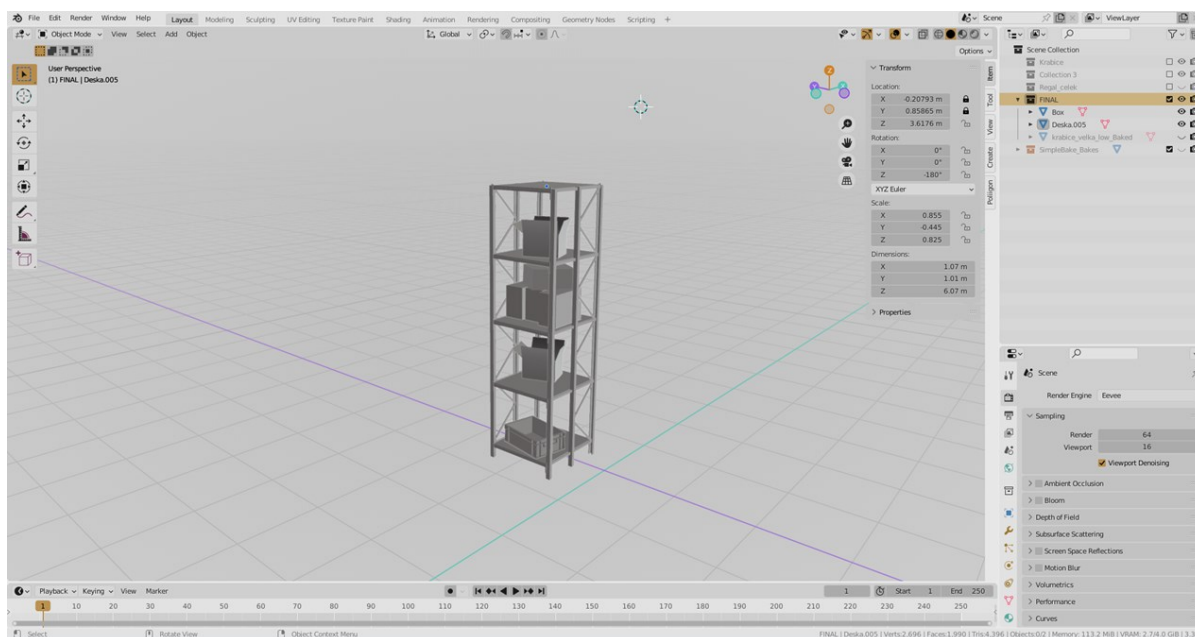
7.2 Tvorba simulátorů

Dalším krokem, bylo přidání UMM generátoru, do pracovní scény, ve které se připraví prefaby veškerých bludišť. Je potřeba vytvořit 9 různých bludišť o 3 různých velikostech a změřit délku správné cesty a určit počet slepých konců. Pro přehlednost celého projektu se veškeré vytvořené komponenty budou ukládat do složky „1“, která se vytvořila v hierarchii projektu. První scéna, je již vytvořená ze založení samotného projektu. Tuto scénu využijeme, pouze ji přejmenujeme pro lepší přehlednost. Pravým tlačítkem klikneme na „SampleScene“ a dáme „Save Scene As“. Nyní se může uložit do složky 1 a pojmenovat „Bludiste“

7.2.1 Vytvoření vzhledu bludiště

Před samotným generováním bylo důležité upravit celkový vzhled bludiště, tak aby se při následném generování, vytvářel již upravený vzhled. Pokud by se vzhled upravoval až po vygenerování, muselo by se každé bludiště znovu přegenerovat s nově vytvořenými prvky pro stěny. Nebo by se musely každý prvek stěny předělávat ručně, což by bylo velmi neefektivní.

Generátor používá k sestavení celého bludiště bloky o rozměrech 1 x 1 metr. Vytvoříme si proto model regálu o stejných rozměrech. Pro tvorbu tohoto modelu využijeme modelovací software Blender. Cílem modelu, který následně využijeme ve VR a obecně ve hrách, je jeho co možná nejmenší grafická náročnost. Toho se dosáhne tím, že se mesh²⁴ modelu bude skládat z co nejméně prvků. Z (Obrázek 35) je vidět, že model má 1 990 prvků.



Obrázek 35 Blender – Model regálu

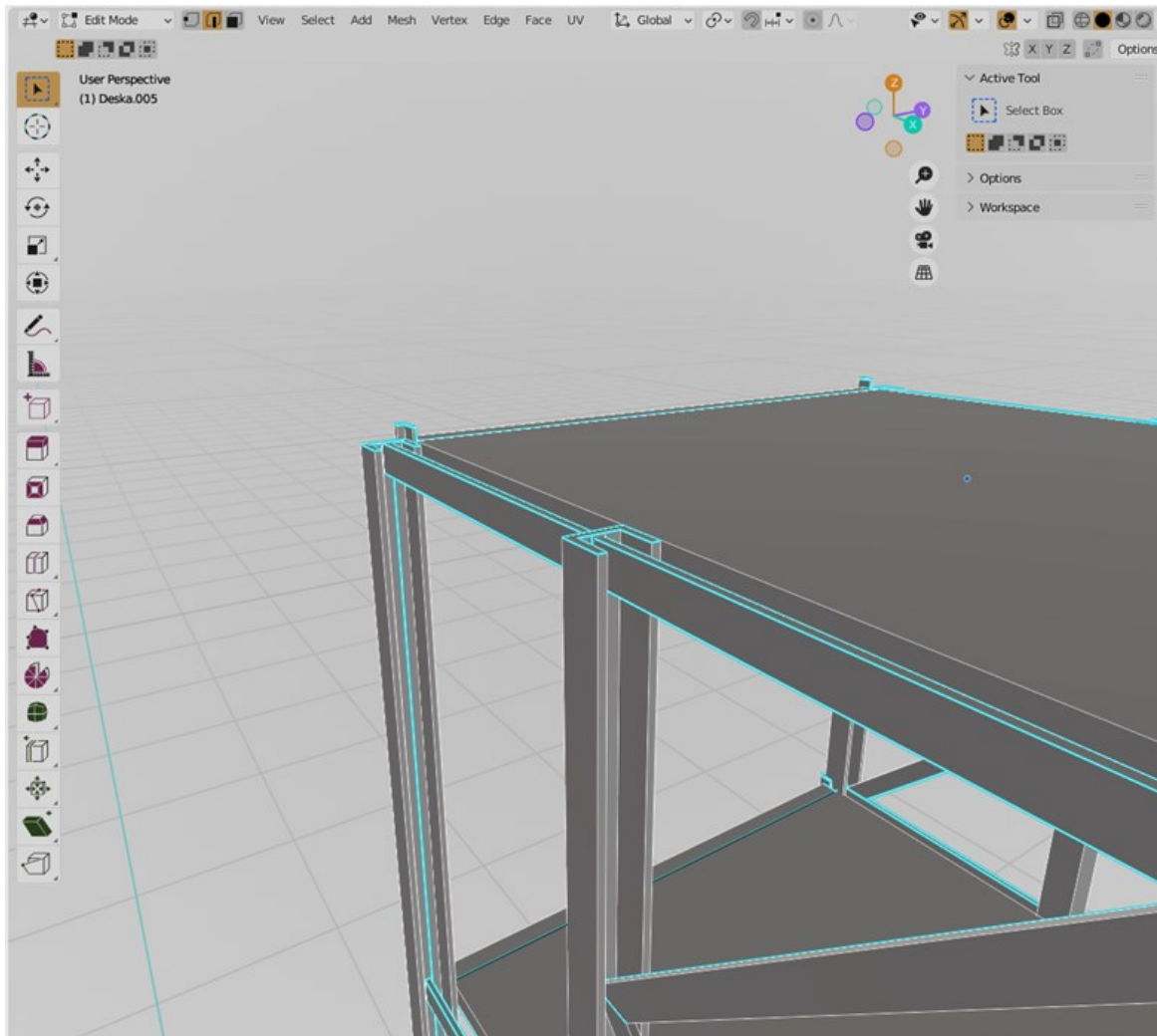
Součástí tvorby modelu, je také vytvoření textur. K tvoření textur jsme opět využili Blender. Blender u textur využívá systém tzv. Nodes v překladu uzly. Uzly v Blenderu jsou klíčovým prvkem při tvorbě textur. Pomocí node editoru lze vytvořit složité texturovací systémy, které umožňují vytvářet realistické a komplexní materiály pro objekty ve scéně. Každý uzel v texturovacím systému může mít vstupy pro textury, barvy, normálové mapy a další datové typy, a také výstupy pro tyto informace, které se používají k vytváření výsledné textury. Uzly mohou být například šumové generátory pro vytváření organických textur, filtru pro úpravu jasu a

²⁴ Mesh – Je soubor vrcholů (vertexů) a hran (edges), které vytvářejí polygonální síť a tvoří tak 3D tvar objektu.

kontrastu, nebo dokonce složitější uzly, které umožňují tvorbu procedurálních textur pomocí matematických funkcí.

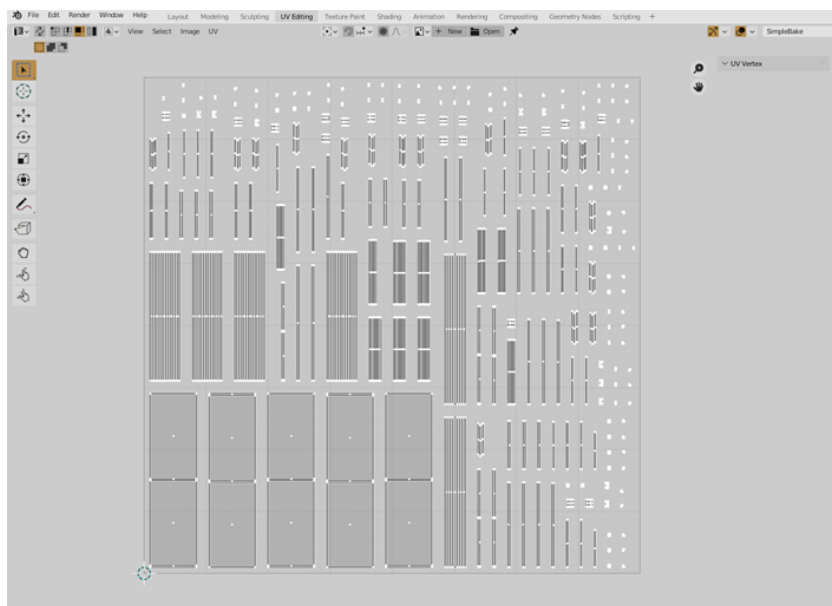
K vytvoření textur v Blenderu je důležité mít správně připravený 3D model. Jednou z klíčových částí této přípravy je tzv. "UV unwrapping", česky nazýváno rozvinutí povrchu modelu. Tento proces slouží k přeměně 3D modelu na 2D plochu, kterou lze poté texturovat. Proces UV unwrappingu spočívá v rozřezání modelu na kusy a následném "rozbalení" těchto kusů na 2D plochu. Tento proces v podstatě vytváří mapu povrchu objektu, která umožňuje aplikovat textury na správná místa na modelu. V Blenderu lze UV unwrapping provést pomocí různých technik, například manuálně ručním rozřezáním modelu, nebo automaticky pomocí nástrojů, jako je například "Smart UV Project".

Pro náš model, který má navazující části pod různými úhly a celkově je složen z disproporčních tvarů, bylo nutné provést UV unwrap ručně. To probíhá tak, že se postupně označí hrany modelu, které mají fungovat, jako tzv. seams (řezy) v textuře. Místa, ve kterých se textury napojují na sebe. Tento proces je poměrně časově náročný, ale nezbytný. Jednotlivé seams, jsou pak v Blenderu vyznačeny modrou barvou.



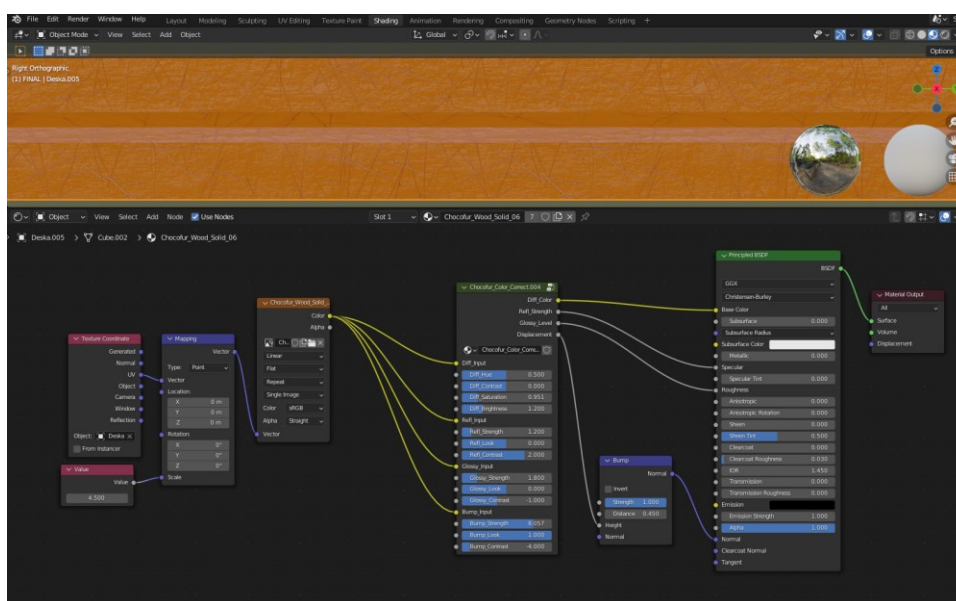
Obrázek 36 UV unwrap

Tímto procesem jsme pak schopni „rozbalit“ 3D model na 2D plochu a následně na tuto plochu nanést textury. Model z (Obrázek 36) je znázorněn ve 2D na (Obrázek 37).



Obrázek 37 UV unwrapnutý model

Na takto připravený model, jsme pomocí nodes editoru mohli vytvořit vhodné materiály. Zde je důležité rozlišovat pojem materiál a textura. Blender totiž pracuje s materiály. Materiály jsou vytvořené pomocí již zmiňovaných nodes. Nicméně jiné softwary, jako například Unity s těmito nodes pracovat neumí. Unity pracuje s texturami. Pro tento případ existuje v Blenderu možnost tzv. bakingu²⁵ (zapékání). Díky tomu jsme schopni materiály, které používáme na modelech převést na textury. Tento proces taktéž vyžaduje správný UV unwrap našeho modelu.



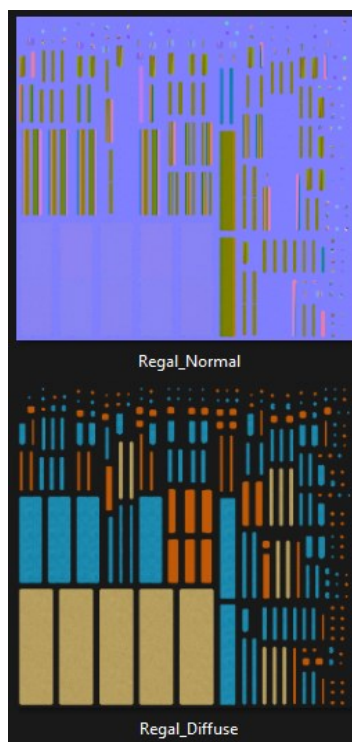
Obrázek 38 Blender – Nodes

Zapečené mapy jsme pak uložili, společně s vyexportovaným modelem ve formátu FBX, se kterým pracuje software Unity. Při jeho exportu je důležité provést správné nastavení pro lépe optimalizovaný export. Z Blenderu jsme dostali výstupy v podobě textur a modelu FBX.

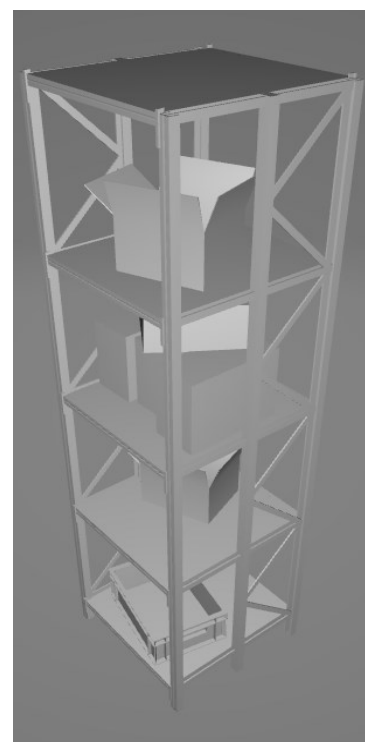
²⁵ Baking – Je zde myšlený, jako proces, při kterém převedeme složitý materiál vytvořený v Blenderu na obrázkový formát (texturu), kterou následně můžeme využít v Unity. Tento pojem lze také využít v souvislosti přenesení detailní textury složitého modelu na model jednodušší a méně graficky náročný.



Obrázek 39 Nastavení exportu FBX



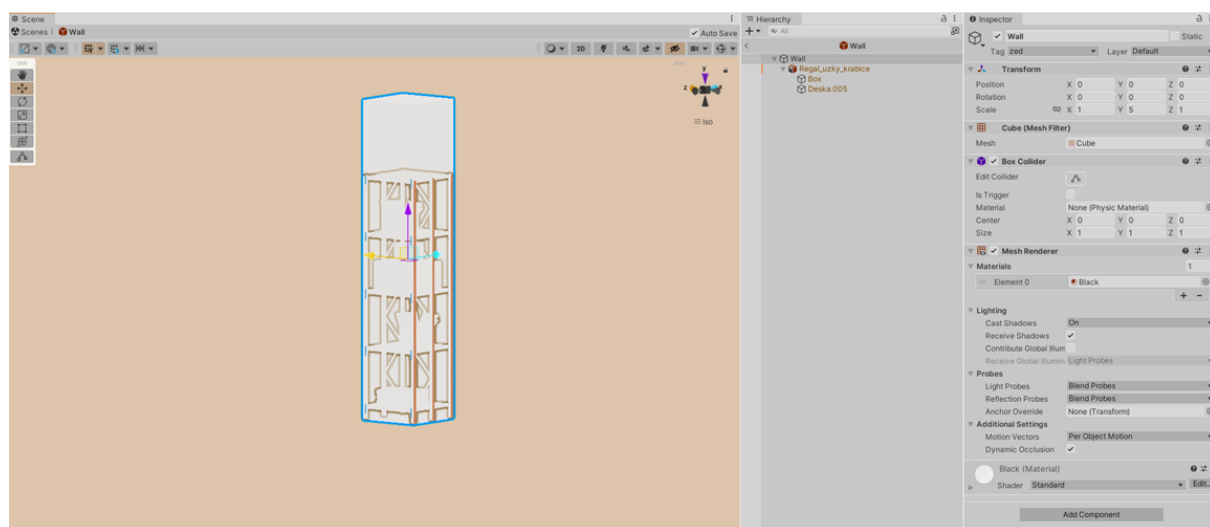
Obrázek 40 Textury



Obrázek 41 FBX model

Pro připravený model, byla v projektu vytvořena složka „modely“. Do této složky se přesunul model společně s připravenými texturami, které se na model automaticky aplikují, pokud jsou přidány společně.

Po přidání modelu do Unity, se přiřadily, k již vytvořenému prefabu zdi, který využívá UMM generátor. Vyvarovali jsme se tak nastavování pivotu na správnou pozici, tak aby při generování bludiště byly zdi ve správné výšce, a převzaly se tak tyto vlastnosti od vyzkoušeného prefabu. Zároveň se pomocí viditelného meshe původní zdi mohlo nastavit správné měřítko mého modelu, tak aby při generování bludiště, nevznikaly mezi jednotlivými bloky mezery viz Obrázek 42.

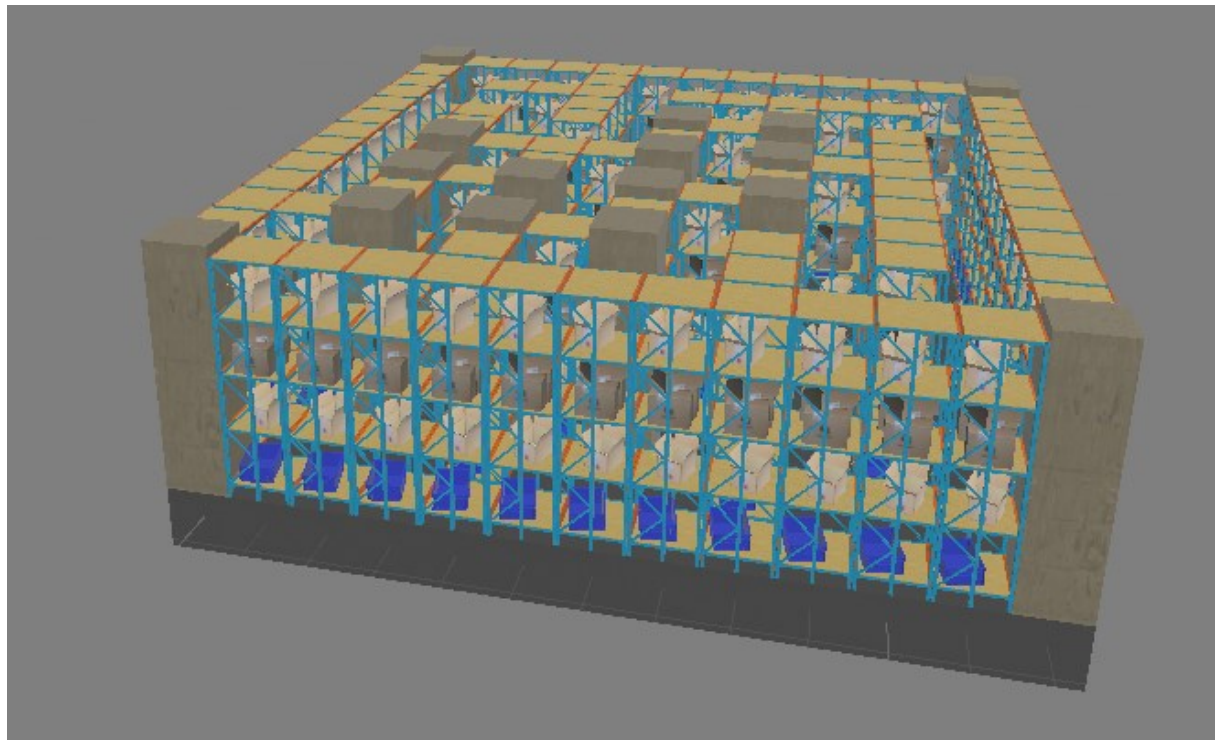


Obrázek 42 Stěna

Po nastavení pozice a měřítka, tak aby model byl souhlasný s původním, stačilo pouze odškrtnout políčko „Mesh Renderer“ a tím zmizel mesh (bílý) původního modelu.

Modely podlahy a sloupů, které vystupují na rozích byly zanechány původní. Pouze se jim změnil materiál, tak aby více odpovídal industriálnímu skladu.

Připravené prefaby se přesunuly do jim určených políček v UMM editoru viz Obrázek 28.

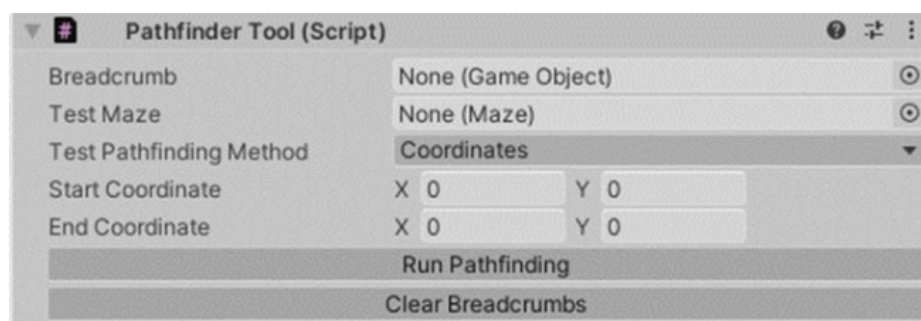


Obrázek 43 Upravený vzhled bludiště

7.2.2 Hledání správné cesty

Před samotným generováním bludiště je důležité popsat, jak funguje doplněk UMM generátoru PathFinding Tool (dále jen PFT). Tento doplněk byl využit pro hledání správné cesty v bludišti a následné určení délky této cesty.

Pro přidání editoru bylo nutné vytvořit prázdný GameObject, který byl pojmenován „Finder“ a následně se mu přes tlačítko „Add Component“ přiřadil skript²⁶ „PathFinder Tool“.

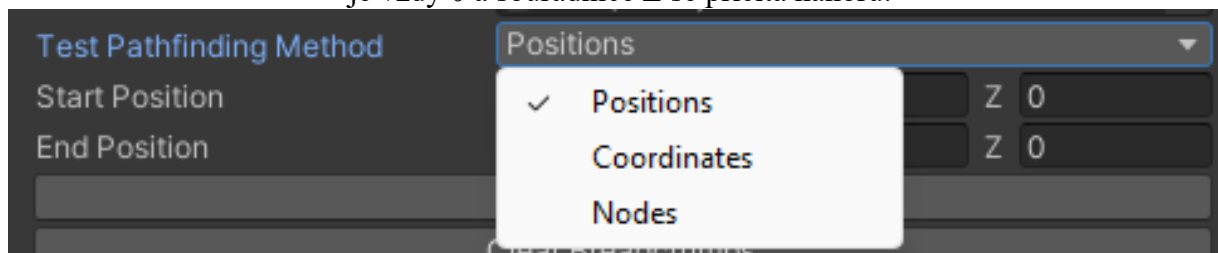


Obrázek 44 PathFinding Tool – Editor

Tím se nám otevře editor PFT. Tento editor obsahuje následující nastavení:

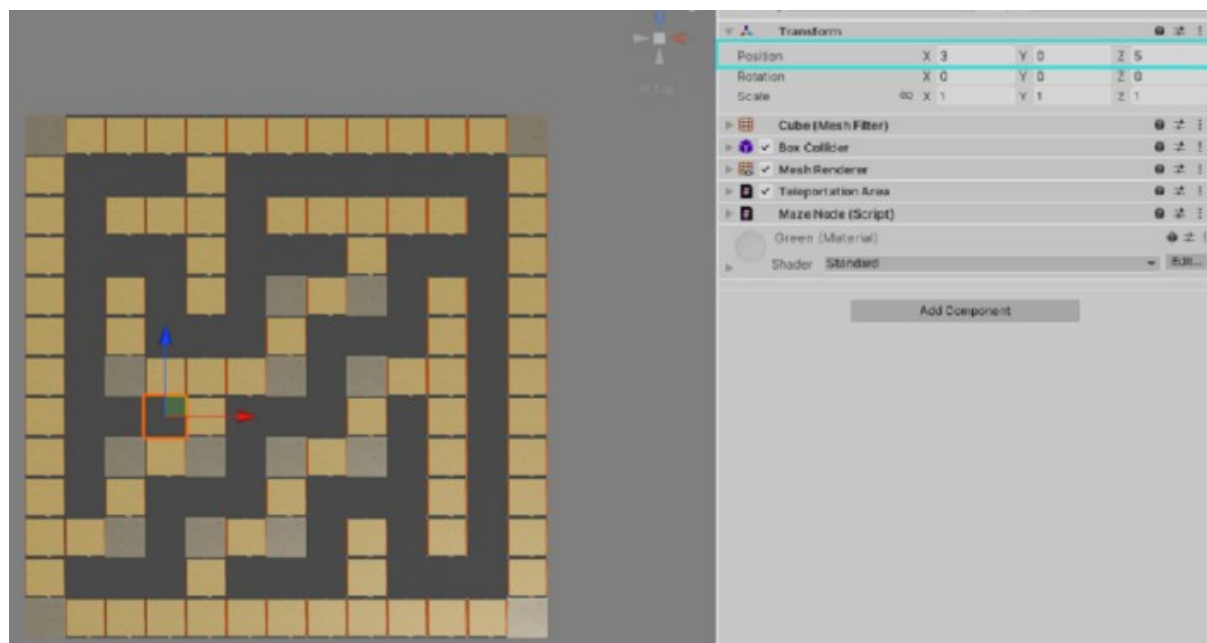
²⁶ Skript – Je soubor napsaný v jazyce C#, který obsahuje instrukce a funkce pro ovládání a interakci s herním objektem nebo scénou.

- Breadcrumb* – Pole, do kterého se umísťuje objekt, kterým zvýrazníme cestu. Využijeme již vytvořený prefab šipky přímo z knihovny UMM generátoru.
- Test Maze* – Zde se umísťuje vygenerované bludiště, na které chceme skript aplikovat.
- Test Pathfinding Method* – Výběr metody, kterou chceme použít pro určení startu a cíle v bludišti. Využijeme metodu Positions. Pozice bloků v bludišti jsou určeny v souřadnicovém systému bludiště. Levý dolní roh má pozici X:0 Y:0 Z:0. Souřadnice X se přičítá směrem vpravo, souřadnice Y je vždy 0 a souřadnice Z se přičítá nahoru.



Obrázek 45 Metody hledání cesty

- Start Position* – Nastavení pozice startu bludiště. Tato pozice bude vždy X:1, Y:1, Z:1.
- End Position* – Nastavení pozice startu bludiště. Tuto pozici zjistíme z informace bloku, vygenerovaného bludiště, který si určíme jako cílový.



Obrázek 46 Pozice bloku bludiště

- Run PathFinding* – Tlačítko, kterým vygenerujeme správnou cestu v bludišti.
- Clear Breadcrumbs* – Tlačítko, kterým smažeme zvýraznění správné cesty.

Poslední věc, která v tomto bodě chybí, bylo určení délky cesty. UMM generátor, žádnou takovou funkci nenabízí.

Vytvořil se proto skript, který tuto funkci přidává. Využívá již vytvořených objektů, které cestu znázorňují. Jelikož PFT přiřazuje vždy právě jeden objekt k jednomu poli a seznam těchto

objektů je veden v hierarchii scény pod stejným jménem, tak stačí pouze najít všechny objekty se jménem „Breadcrumb(Clone)“ a vypsat jejich počet do console.

```
// Nastaví se hledané jméno objektu
public string objectName = "Breadcrumb(Clone)";

// Proměnná pro uchování počtu nalezených objektů
private int objectCount;

// Metoda pro spočítání počtu objektů s daným jménem
void StartCounting()
{
    // Najde se všechny objekty v aktuální scéně
    GameObject[] objects = GameObject.FindObjectsOfType<GameObject>();
    // Počet se inicializuje na nulu
    objectCount = 0;
    // Prochází se všechny nalezené objekty
    foreach (GameObject obj in objects)
    {
        // Pokud jméno objektu odpovídá hledanému jménu
        if (obj.name == objectName)
        {
            // Zvýší se počet
            objectCount++;
        }
    }

    // Vypíše se délka správné cesty bludiště (tj. počet nalezených objektů)
    Debug.Log("Délka správné cesty bludiště je: " + objectCount);
}

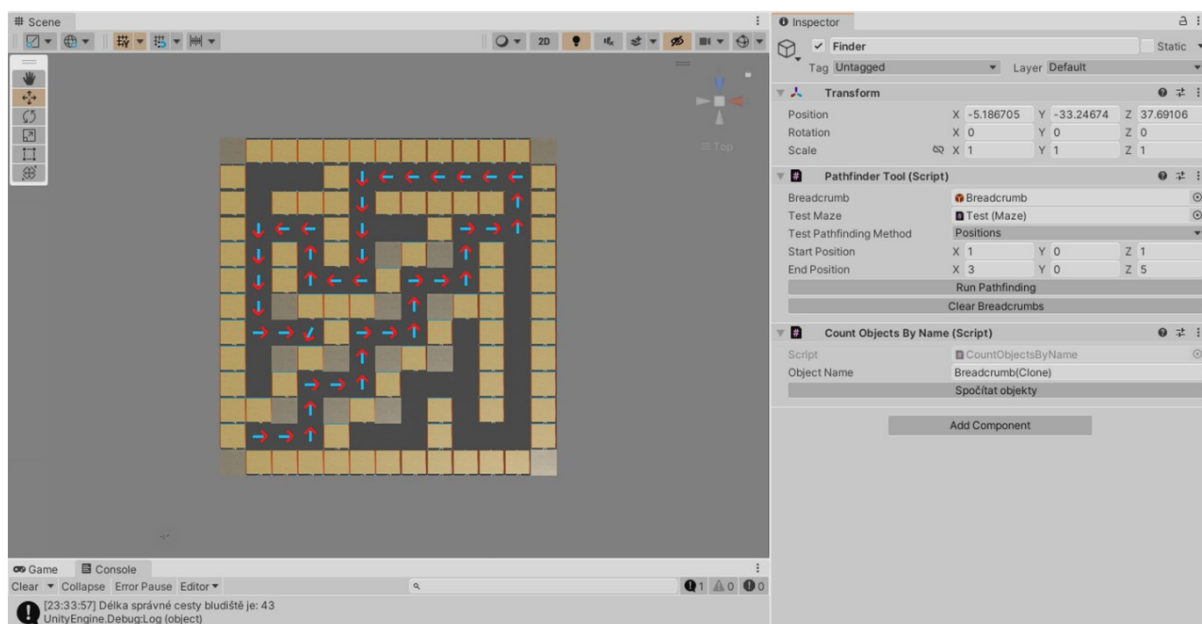
// Metoda pro vynulování počtu objektů
void ClearCount()
{
    objectCount = 0;
}

#if UNITY_EDITOR
[UnityEditor.CustomEditor(typeof(CountObjectsByName))]
public class CountObjectsByNameEditor : UnityEditor.Editor
{
    public override void OnInspectorGUI()
    {
        DrawDefaultInspector();

        // Získá se reference na objekt třídy CountObjectsByName
        CountObjectsByName myScript = (CountObjectsByName)target;

        // Pokud už existují nějaké objekty s daným jménem
        if (myScript.objectCount > 0)
        {
            // Vynuluje se počet
            myScript.ClearCount();
        }

        // Po stisknutí tlačítka "Spočítat objekty" se spustí metoda pro spočítání objektů
        if (GUILayout.Button("Spočítat objekty"))
        {
            myScript.StartCounting();
        }
    }
}
#endif
```



Obrázek 47 Generování bludiště

7.2.3 Generování bludišť

Po celkovém nastavení UMM editoru, vzhledu bludiště a připravení PFT společně s komponentou pro zjištění délky správné cesty, přišlo na řadu jejich generování.

Nejdříve se začala generovat bludiště úrovně I o rozměrech 11 x 11. Cílem bylo vygenerovat 3 bludiště o stejné velikosti, stejném počtu slepých cest a stejné délce správné cesty viz Tabulka 3.

Postup generování:

- 1) V UMM editoru se nastavil rozměr bludiště na 11 x 11.
- 2) Generovalo se bludiště o náhodném seedu.
- 3) Subjektivně jsme určili konec bludiště.
- 4) Pomocí PFT se zjistila správná cesta a její délka.
- 5) Ručně se spočítal počet slepých konců.
- 6) Tyto údaje se zapsaly společně se seedem do tabulky.
- 7) Pokračovalo se v generování bludišť, dokud se v tabulce nedostali 3 bludiště o stejných parametrech.

Vzhled výsledných bludišť pro I. kategorii:

Pro generování bludišť II. kategorie, bylo třeba přenastavit velikost na 15x15. V druhém kroku se přestaly generovat kompletně náhodná čísla seedů, ale zvyšovala se o 1.

Vzhledem k zvětšení bludiště se zvětšil možný počet konců. Tudíž kroky 3–6 se mohly opakovat pro jedno bludiště víckrát, v souladu s počtem možných konců. Tato změna je zachycena v Tabulka 4.

Pro kategorií III. se postupovalo stejně, jen došlo ke změně u druhého kroku. Volení seedu probíhalo od jedničky viz Tabulka 5.

Tabulka 3 Bludiště I

I				
Pořadí	Seed	Počet kroků		Slepý konce
7.	879119668	15		3 43
14.	-1049608006	15		3 43
19.	795360431	15		3 43

Tabulka 4 Bludiště II

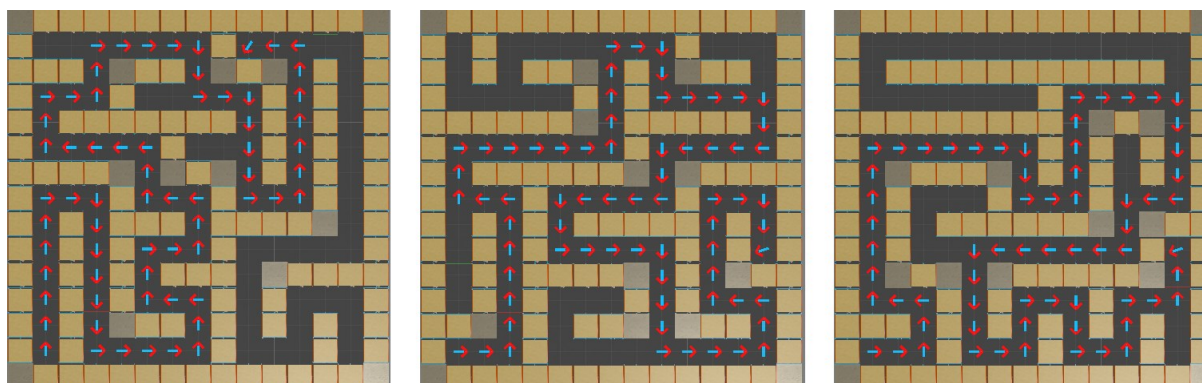
II							
Pořadí	Seed	Počet kroků		Slepý konce		Délka cesty	
13.	1141355220	–	22	–	4	–	65
30.	1141355237	20	22	2	4	–	65
36.	1141355243	–	22	–	4	–	69

Tabulka 5 Bludiště III

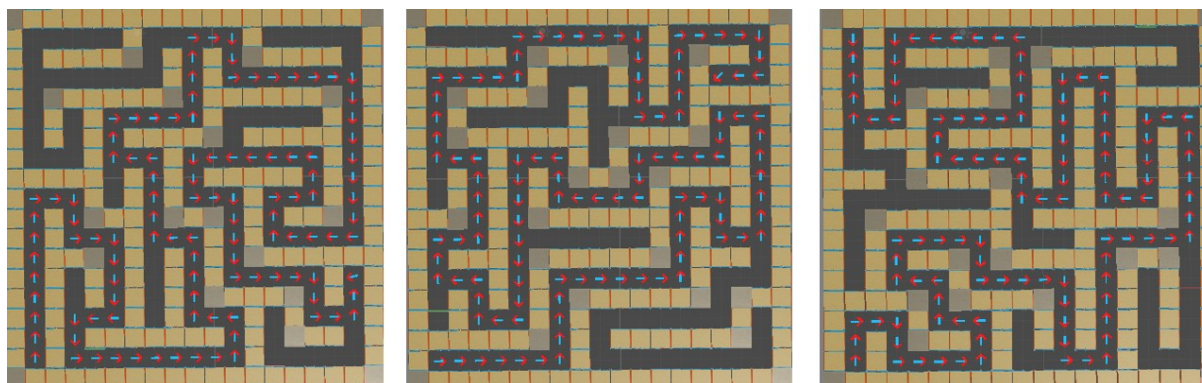
III													
Pořadí	Seed	Počet kroků				Slepý konce				Délka cesty			
6.	6	–	30	32	33	–	–	–	7	–	–	97	109
19.	19	–	–	33	35	–	–	–	7	–	–	109	–
21.	21	–	–	32	33	–	–	4/5	6	–	–	–	109

Celková data tabulek 3 až 5 v podobě digitální přílohy.

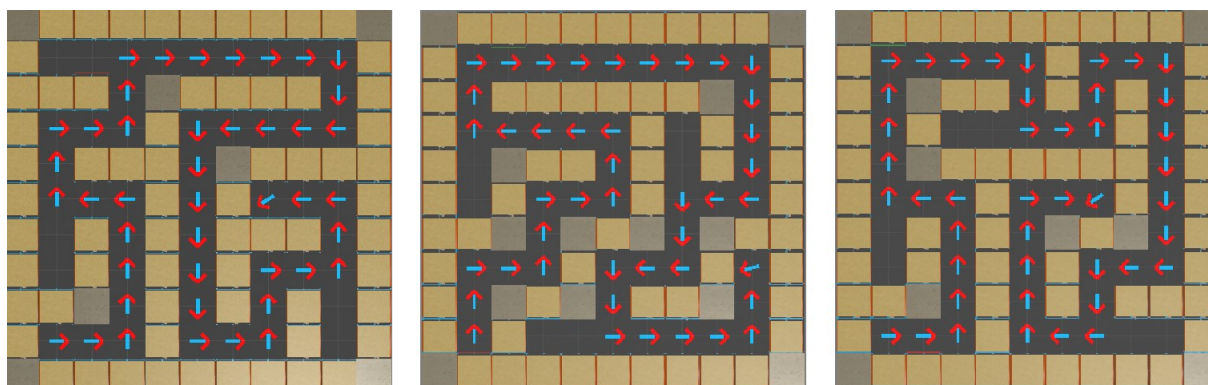
Vzhled výsledných bludišť pro kategorii I, II. a III.:



Obrázek 48 Kategorie II



Obrázek 49 Kategorie III



Obrázek 50 Kategorie I

Finální parametry všech kategorií:

Tabulka 6 Parametry testovacích prostředí

Parametry testovacích prostředí			
Kategorie	I	II	III
Velikost	11x11	15x15	19x19
Počet kroků	15	22	33
Počet slepých konců	3	4	7,6
Délka cesty	43	67+-2	109

Jak je z Tabulka 6 patrné, povedlo se určit stejné parametry pro I. Kategorii. V II. kategorii se liší délka cesty o +/- 2 metry, tento rozdíl je vzhledem k celkové délce a tvaru tras zanedbatelný. U III. kategorie má jedno z bludišť o jeden slepý konec navíc. Jak si můžeme všimnout jistě záludnosti určitých slepých konců, tak je tento rozdíl taktéž zanedbatelný.

Posledním krokem byla optimalizace všech bludišť. Podlaha, která byla vygenerována UMM generátorem se skládá z jednotlivých bloků podlahy. Tyto bloky byly využity při hledání a určování délky cest. Ve fázi testování by však mohly jejich nadbytečné polygony narušovat chod aplikace. Proto se všechny objekty podlahy u všech bludišť smazaly a nahradily se jedním objektem „plane“, který můžeme vytvořit přímo ve Scene View bludiště a použili jsme ho jako podlahu. Tímto krokem se výrazně snížila náročnost aplikace na výkon.

Všechna bludiště, byla uložena jako prefaby do složky projektu „Bludiste“.

7.2.4 Skripty

Pro finální funkčnost VR prostředí bylo zapotřebí vytvořit skripty, které se starají o určité funkcionality ve výsledné aplikaci. Veškeré skripty byly vytvořeny ve [Visual Studiu 2019 verze 16.11.25](#).

Funkčnost aplikace

Následující skripty umožňují správnou funkčnost aplikace napříč všemi ovládaními.

Dalším skriptem, který v aplikaci využijeme, je skript „Casovac“. Tento skript slouží k odpočítávání času, za který je hráč schopný projít bludištěm. V momentě, kdy projde objektem start, spustí se časomíra, která běží na pozadí. Po průchodu hráče cílovým objektem se časomíra zastaví a na obrazovce se ukáže výsledný čas. Tento čas pak uživatel nahlásí. Zobrazení času na obrazovce bylo využito z důvodu nenáročnosti propojování komunikace mezi brýlemi a vzdáleným počítačem, do kterého by se hodnoty musely ukládat. Skript zobrazuje čas v textovém poli ve formátu MM:SS:FF (minuty, sekundy, setiny) viz Obrázek 51.

```
public float cas = 0f; // Aktuální čas od začátku hry
public Text casText; // Textové pole pro zobrazení času
public GameObject startObjekt; // Startovní objekt, který spouští odpočítávání času
public GameObject cilObjekt; // Cílový objekt, který ukončuje odpočítávání času

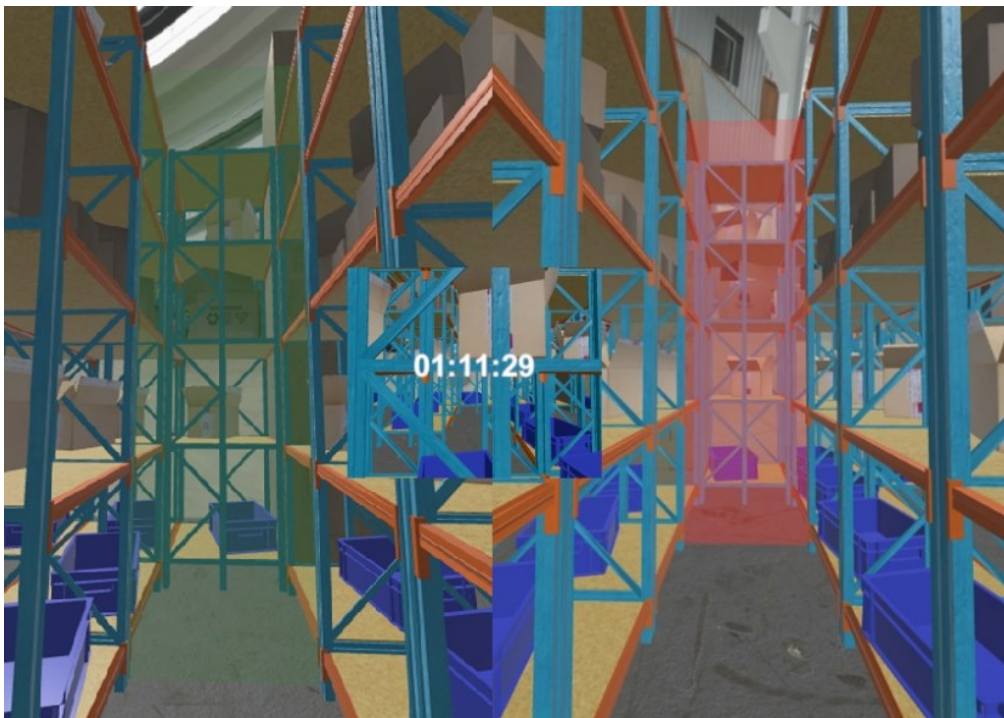
private bool start = false; // Zda hráč prošel startovní objekt
private bool cil = false; // Zda hráč prošel cílový objekt

private void Update()
{
    // Pokud hráč pošel startem, ale ještě ne cílem
    if (start && !cil)
    {
        cas += Time.deltaTime; // Přičíst uplynulý čas od poslední aktualizace
        int minuty = Mathf.FloorToInt(cas / 60); // počet minut z celkového času
        int vteriny = Mathf.FloorToInt(cas % 60); // počet sekund
        float setiny = Mathf.FloorToInt((cas - Mathf.FloorToInt(cas)) * 1000); // počet setin

        // Zobrazit čas v textovém poli ve formátu MM:SS:FFF (minuty, sekundy, setiny)
        casText.text = minuty.ToString("00") + ":" + vteriny.ToString("00") + ":" +
setiny.ToString("000");
    }
}

private void OnTriggerEnter(Collider other)
{
    // Pokud hráč projde startem, nastavit proměnnou start na true
    if (other.gameObject == startObjekt)
    {
        start = true;
    }
    // Pokud hráč projde cílem, nastavit proměnnou cil na true a zobrazit textové pole s časem
    else if (other.gameObject == cilObjekt)
    {
        cil = true;
        casText.enabled = true;
    }
}
}
```

Časovač pak přidáme přímo k OVRCameraRig.



Obrázek 51 Start a cíl

Další skript, který využijeme bude „Centr“. Tento skript pokaždé přesune uživatele na správný startovní bod. Využívá k tomu souřadnic již předem nastaveného objektu na správném místě.

Přidáme ho tedy jako komponentu na objekt, který má fungovat jako centrovač. Tento objekt je pak ve scéně nastaven na startovní bod (pole startu v bludišti).

```
// Zadávání úhlu natočení kamery po přemístění
public int degree;
// Čas do teleportace
public float resetPositionTime = 0.1f;
public bool playOnAwake;
void Awake()
{
    Invoke("LateStart", 0.1f);
    if (playOnAwake == true)
    {
        Invoke("ResetVRPosition", resetPositionTime);
    }
}
public void LateStart()
{
    cameraDefaultYValue = m_CameraRig.position.y; // Uložit výchozí výšku kamery
}
public void ResetVRPosition()
{
    float currentRotY = m_CentreEyeAnchor.eulerAngles.y; // Získat aktuální rotaci kamery
    float targetYRotation = 0.0f; // Cílová rotace kamery
    float difference = targetYRotation - currentRotY + degree; // Vypočítat rozdíl rotace
    m_CameraRig.Rotate(0, difference, 0); // Natočit kameru podle vypočítaného úhlu
    Vector3 newPos = new Vector3(this.transform.position.x - m_CentreEyeAnchor.position.x, 0,
this.transform.position.z - m_CentreEyeAnchor.position.z); //Vypočítat novou pozici kamery
    m_CameraRig.transform.position += newPos; // Přesunout kameru na novou pozici
    m_CameraRig.transform.parent = this.gameObject.transform; // Nastavit rodiče kamery na tento objekt
    cameraY = cameraDefaultYValue - m_CentreEyeAnchor.localPosition.y; // Vypočítat výšku kamery
    m_CameraRig.localPosition = new Vector3(m_CameraRig.localPosition.x, cameraY,
m_CameraRig.localPosition.z); // Nastavit novou pozici kamery
}
```

Ovládání rukama

Následující skripty bylo nutné přidat pro správnou funkčnost ovládání za pomoci gest rukou.

Prvním je „zatemnovac“. Jedná se o skript, který uživateli zatemní kameru, pokud by v prostředí bludiště vešel do stěny. Tento skript bude umístěn na objekt v naší scéně a využijeme ho při ovládání pomocí gest rukou, kdy nemáme možnost omezit pohyb hráče.

Zatemňovač umístíme na objekt „koule“ která je přímo vázána s kamerou hráče. Tuto kameru získáme ze souborů pluginu Oculus Integration pod názvem OVRCameraRig. Po rozbalení tohoto prefabu umístíme objekt „koule“ v hierarchii pod CenterEyeAnchor. Tím se sváže zatemňovací koule s úrovní hlavy uživatele s VR headsetem.

```
public class Zatemnovac : MonoBehaviour
{
    private int numColliding = 0;
    private Renderer objectRenderer;

    private void Start()
    {
        // Získání komponenty Renderer objektu, ke kterému je tento skript připojen
        objectRenderer = GetComponent<Renderer>();
    }
    private void OnTriggerEnter(Collider other)
    {
        // Zkontrolování, zda kolidující objekt má tag "zed"
        if (other.CompareTag("zed"))
        {
            numColliding++;
            // Pokud se objekt poprvé dotýká objektu s tagem "zed", spustit postupné zneprůhlednění
            if (numColliding == 1)
            {
                StartCoroutine(FadeObject(true, 1.0f));
            }
        }
    }
    private void OnTriggerExit(Collider other)
    {
        // Zkontrolování, zda kolidující objekt má tag "zed"
        if (other.CompareTag("zed"))
        {
            numColliding--;

            // Pokud objekt již nekoliduje s žádným objektem s tagem "zed", spustit postupné zprůhlednění
            if (numColliding == 0)
            {
                StartCoroutine(FadeObject(false, 1.0f));
            }
        }
    }
    private IEnumerator FadeObject(bool fadeIn, float duration)
    {
        // Získání počáteční hodnoty průhlednosti objektu
        float startAlpha = fadeIn ? 0.0f : 1.0f;
        float endAlpha = fadeIn ? 1.0f : 0.0f;
        // Postupné zvýšení nebo snížení průhlednosti objektu v průběhu času
        float elapsedTime = 0.0f;
        Color objectColor = objectRenderer.material.color;
        while (elapsedTime < duration)
        {
            float t = elapsedTime / duration;
            objectColor.a = Mathf.Lerp(startAlpha, endAlpha, t);
            objectRenderer.material.color = objectColor;

            elapsedTime += Time.deltaTime;
            yield return null;
        }
        // Nastavení konečné hodnoty průhlednosti objektu
        objectColor.a = endAlpha;
        objectRenderer.material.color = objectColor;
    }
}
```

Pro ovládání za pomoci gest rukou, byl zvolen způsob pohybu přes teleport. Hráč tedy pravou rukou míří na pozici, kam se chce přemístit a gestem druhé ruky se na tuto pozici přemístí. Gesto pro teleportaci bylo zvoleno sevření pěsti. Umístily se tedy dva collidery. První na střed dlaně, který se dá najít v hierarchii prefabu „OVRCustomHandPrefab_L“ a druhý na poslední článek prostředníčku levé ruky. Aby bylo možné hráče na danou pozici přemístit, bylo zapotřebí vytvořit funkci, která nám tuto pozici bude určovat. K tomu se využila komponenta „Spline“. Díky této komponentě a následujícímu skriptu, jsme schopni zjistit souřadnice, ve které se paprsek dotýká podlahy a umístit do tohoto bodu objekt „Teleport Mark“ ke kterému se pak hráč přemístí.

```
public GameObject Hand; // Odkaz na ruku hráče („OVRCustomHandPrefab_L“)
public GameObject TeleportMark; // Odkaz na teleportační značku „Koule“
public SplineComputer sc; // Odkaz na spline pro vytváření křivek
private SplinePoint[] points; // Pole bodů křivky
public GameObject Spline; // Odkaz na objekt spline
public float Angle = 0f; // Úhel

void Start()
{
    var c = 3; // Počet bodů křivky
    points = new SplinePoint[c]; // Inicializace pole bodů

    for (int i = 0; i < c; i++)
    {
        points[i] = new SplinePoint();
        points[i].position = Hand.transform.position; // Nastavení pozice bodu na pozici ruky hráče
        points[i].normal = Vector3.up; // Nastavení normály na směr vzhůru
        points[i].size = 0.3f; // Velikost bodu
        points[i].color = Color.white; // Barva bodu
    }
    sc.SetPoints(points); // Nastavení bodů křivky do splinu
}

// Metoda CalculateShiftedPoint, která vrací nový bod, který je posunutý ve směru osy Y o hodnotu shift vzhle-
dem k úsečce mezi bodem pointA a bodem pointB
public Vector3 CalculateShiftedPoint(Vector3 pointA, Vector3 pointB, float shift)
{
    Vector3 lineDirection = (pointB - pointA).normalized;
    Vector3 pointC = pointA + lineDirection * Vector3.Dot(pointB - pointA, lineDirection);

    return new Vector3(pointC.x, pointC.y + shift, pointC.z);
}

void Update()
{
    RaycastHit hit;

    if (Physics.Raycast(Hand.transform.position, Hand.transform.TransformDirection(Vector3.down), out hit,
    Mathf.Infinity))
    {
        if (hit.transform.tag == "Floor") // Kontrola, zda kolizní objekt má tag "Floor"
        {
            Spline.SetActive(true); // Aktivace objektu spline
            TeleportMark.SetActive(true); // Aktivace teleportační značky

            Debug.DrawRay(Hand.transform.position, Hand.transform.TransformDirection(Vector3.down) *
            hit.distance, Color.yellow);

            TeleportMark.transform.position = hit.point; // Nastavení pozice teleportační značky na místo zá-
            sahu raycastem
            points[0].position = Hand.transform.position; // Nastavení pozice prvního bodu křivky na pozici
            hráče
            points[1].position = CalculateShiftedPoint(Hand.transform.position, hit.point, 1f); // Výpočet a
            nastavení pozice druhého bodu křivky
            points[2].position = hit.point; // Nastavení pozice třetího bodu křivky na místo zásahu raycastem
            sc.SetPoints(points); // Nastavení bodů křivky do splinu
        }
    }
    Else
    {
        Spline.SetActive(false); // Deaktivace objektu spline
        TeleportMark.SetActive(true); // Aktivace teleportační značky
        Debug.DrawRay(Hand.transform.position, Hand.transform.TransformDirection(Vector3.down) * 1000,
        Color.white);
    }
}
```


Tím získáme pozici, na kterou se má hráč přesunout. Zbývá už jen skript pro samotnou funkci přesouvání.

```
public Transform Player; // Souřadnice hráče
public Transform TeleportSpot; // Souřadnice bodu na konci paprsku

private void OnTriggerEnter(Collider other)
{
    if (other.tag == "palecl") // Kontrola, zda došlo ke kolizi. V tomto případě je tag „palecl“ umístěn na prostředek dlaně. K teleportaci tedy dojde pokud uživatel sevře pěst.
    {
        // Přenesení pozici hráče na souřadnice teleportačního místa, zachováva původní výšku hráče
        Player.position = new Vector3(TeleportSpot.position.x, Player.position.y, TeleportSpot.position.z);
    }
}
```

Ovládání ovladači a Cybershoes

Při pohybu ovladači nebo Cybershoes, jsme pak schopni regulovat kolize hráče a stěny tak, že jimi hráč nebude schopný procházet. Toho docílíme aplikací již vytvořených komponent, které simulují fyziku. Tyto komponenty jsou součástí engine unity. Na veškerých stěnách je již z UMM generátoru umístěna komponenta „Box Collider“, která slouží k detekci kolizi s jinými objekty. Do hierarchie objektu „OVRCameraRig“ pak přidáme objekt s názvem „body“ na který umístíme komponentu „Capsule Collider“. Tím vznikne interakce mezi stěnami a hráčem, tak že jimi hráč nebude schopný projít. Je zapotřebí tuto funkci však doplnit o jednoduchý skript, který sváže souřadnice kamery se souřadnicemi naší „body“ komponenty. Bez něj by docházelo k odsouvání kamery mimo kolizní tělo. Tento skript je pak umístěn na komponentu „body“ a do proměnné „head“ umístíme objekt „CenterEyeAnchor“.

```
public Transform head;
void Update()
{
    gameObject.transform.position = new Vector3(head.position.x, head.position.z);
}
```

Pro pohyb ovladači, bylo nastaveno otáčení na levý joystick ovladače a chůze na pravý. Otáčení pak zajišťuje následující skript:

```
public Rigidbody player;
public Transform rotator;
public float angle; // úhel otáčení
private bool rotated = false; // zda-li byl hráč již otočen

void Update()
{
    var ovladacAxis = OVRInput.Get(OVRInput.Axis2D.PrimaryThumbstick, OVRInput.Controller.LTouch);

    if (ovladacAxis.x >= .8f && !rotated) // pokud je páčka daná doprava a hráč nebyl otočen
    {
        player.transform.RotateAround(rotator.position, rotator.up, angle);
        rotated = true; // označení, že hráč byl již otočen
    }
    else if (ovladacAxis.x <= -.8f && !rotated) // pokud je páčka daná doleva a hráč nebyl
    otočen
    {
        player.transform.RotateAround(rotator.position, rotator.up, -angle);
        rotated = true; // označení, že hráč byl již otočen
    }
    else if (ovladacAxis.x < .8f && ovladacAxis.x > -.8f) // pokud je páčka vrácena na střed
    {
        rotated = false; // resetování označení otočení hráče
    }
}
```

Pohyb pak zajišťuje tento skript:

```
public Rigidbody player; // Odkaz na Rigidbody hráče
public float speed; // Rychlost pohybu

void Update()
{
    // Získání hodnoty osy ovladače pravého palce
    var ovladacAxis = OVRInput.Get(OVRInput.RawAxis2D.RThumbstick, OVRInput.Controller.RTouch);

    float fixedY = player.position.y; // Uložení původní výšky hráče

    // Pohyb hráče na základě hodnoty osy ovladače pravého palce a rychlosti
    player.position += (transform.right * ovladacAxis.x + transform.forward * ovladacAxis.y) *
Time.deltaTime * speed;

    player.position = new Vector3(player.position.x, fixedY, player.position.z); // Nastavení původní
výšky hráče v ose Y
}
```

Pohyb pomocí Cybershoes nebylo zapotřebí skriptovat. Využili jsme již vytvořeného skriptu, který je součástí volně dostupného projektu na serveru GitHub. V tomto projektu byl již také nastavený GameObject, který reprezentuje hráče, pro platformu Steam, tedy ekvivalent OVRCameraRigu. Doplnili jsme ho tedy pouze o skript časovač. [32]

7.2.5 VR funkčnost

Závěrem přípravy aplikace je nastavení scén, tak aby po spuštění správně fungovala.

Každá scéna musí obsahovat tyto položky:

- 1) OVRCameraRig²⁷
Obsahuje několik prvků, včetně kamery, ovladačů pro pohyb a nastavení, a dalších prvků, které umožňují interakci s virtuálním prostředím. Jak bylo zmíněno v kapitole 7.2.4, zároveň je k tomuto prefabu přiřazen objekt „koule“ se skripty. Pro jednotlivá ovládání se prefab liší pouze v modelech ovladačů. V případě ovládání rukou, do něj budou přidány modely rukou, které se opět získají z knihovny pluginu Oculus Integration. Pro ovládání za pomoci ovladačů a se tyto modely změny na modely ovladačů viz Obrázek 54.
- 2) Centrovac
Objekt, na kterém je umístěn skript „centrovac“ z kapitoly 7.2.4.
- 3) Model bludiště

Toto jsou nezbytné položky, které musí všechny scény obsahovat. Dále můžeme scénu doplnit o objekt „Directional light“. Tento objekt je přímo z knihovny Unity a prosvětluje scénu.

Pro ovládání za pomoci rukou a ovladačů budeme exportovat aplikace přímo do VR headsetu. Vytvoříme tak 2 scény, které budou rozděleny dle kategorie bludiště. Každá scéna pak bude obsahovat výše uvedené objekty, kdy model bludiště je přiřazen podle příslušné úrovně. Scény byly exportovány zvlášť, jako samostatné aplikace.

Aplikace je třeba uložit ve formátu .apk. Kvůli tomu bylo provedeno nastavení v kapitole 7.1.2. Vytvoření aplikace se provádí přes tlačítko „Build“ viz Obrázek 29. Soubor s formátem APK se pak do brýlí nahraje přes aplikaci SideQuest. Po spuštění aplikace se propojí brýle s počítačem pomocí kabelu a zapne se funkce Quest Link přímo v brýlích. Aplikace brýle sama sparuje. Poté můžeme v menu zvolit tlačítko „Instal APK file from folder on computer“, vybrat vytvořený soubor z Unity a vyexportovat ho do brýlí.

²⁷ Pro scénu určenou k Cybershoes, je „OVRCameraRig“ nahrazen objektem „Player“, který obsahuje veškeré komponenty potřebné pro funkci Valve Index headsetu a Cybershoes.

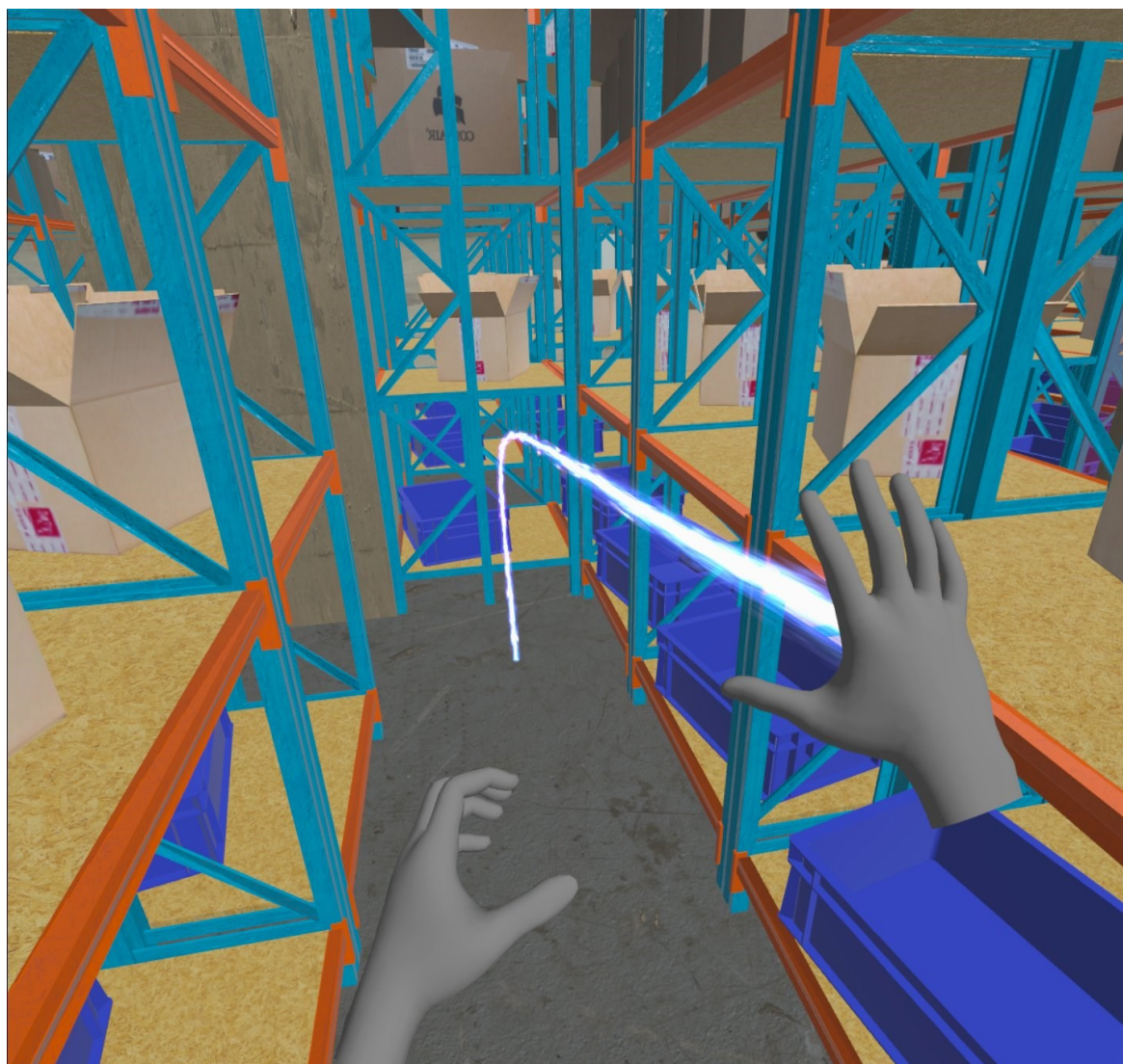


Obrázek 52 SideQuest

Pro ovládání za pomoci Cybershoes se propojí headset přímo s počítačem, na kterém bude aplikace spuštěná přímo v Unity. K propojení počítače a headsetu pak dochází pomocí kabelu. Zároveň je však nutné mít na počítači zapnutou aplikaci SteamVR, která mu dovolí komunikovat s Valve Index headsetem. Ke komunikaci aplikace v unity a headsetu pak slouží již zmiňovaný SteamVR plugin, který jsme si do projektu přidali.

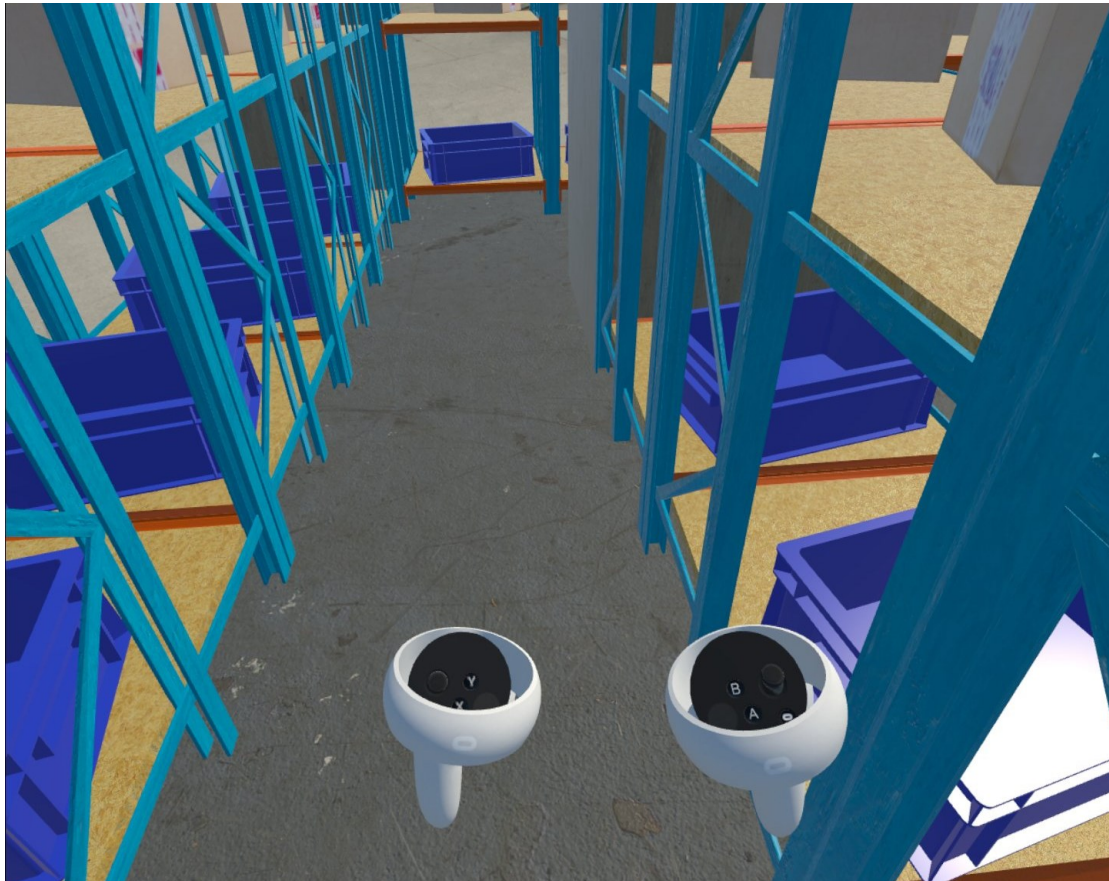
7.3 Shrnutí

Jsou tak připravené tři ovládání. Prvním je ovládání za pomoci gest rukou, zde jedna ruka ovládá pozici teleportačního paprsku a druhou se pak provádí gesto pro teleportování.



Obrázek 53 Teleport

Při ovládání ovladači je pak na pravé páčce nastaven pohyb. Tedy pohyb hráče kopíruje pohyb páčky ovladače, kterou hráč ovládá pravým palcem. Otáčení pak nastavené na páčku levého ovladače, kdy při pohybu páčky vlevo či vpravo se hráč otočí stejným směrem o 45°. Rychlost pohybu zde byla nastavena na 1.7, což odpovídá skutečné rychlosti reálné chůze.



Obrázek 54 Pohyb ovladači

U ovládání využitím Cybershoes uživatel žádné ovladače nevidí, jelikož v tomto případě nejsou potřeba. K pohybu pak dochází posuvem nohou po koberci a otáčení je pak řízeno otáčením uživatele na židli. Rychlost pohybu byla nastavena na 1.7, stejně jako tomu je u ovladačů.



Obrázek 55 Pohyb Cybershoes

8 Ověření simulátorů

Testování se zaměřuje na porovnání orientace ve VR s těmito ovládními, rychlosti průchodu bludištěm a celkovým uživatelským zážitkem, včetně působení virtuální nevolnosti.

Hlavní výzkumné otázky:

- Která varianta ovládní povede k nejrychlejšímu průchodu?
- Která varianta ovládní bude uživatelsky nejpřívětivější?
- V jaké míře bude vnímaná virtuální nevolnost?

8.1 Metodika testování

Testování se rozdělí do 3 základních kroků, které jsou níže popsány. Každý účastník těmito kroky projde ve stejném pořadí a nepřerušovaně. Doba trvání celého testu je odhadnuta na 30 minut.

8.1.1 Úvodní část

Tato část, obsahuje 2 první kroky, které slouží s obecným seznámením probanda s cílem práce a průběhem testování.

1) Seznámení s cílem práce

Probandům se objasní cíl práce, kterým je otestování tří konkrétních druhů ovládní VR, ovladačů, gest rukou a Cybershoes. Bude jim vysvětleno, že cílem je porovnat, jak se jim daří se s jednotlivými ovládními orientovat ve VR, jak rychle dokážou projít bludištěm, jaké byly jejich celkové zkušenosti s funkčností daného ovládní, které budou na konci testování zaznamenány prostřednictvím dotazníku společně s jejich vnímáním cybersickness.

2) Vysvětlení technických náležitosti testování

Probandům bude vysvětleno, jaký bude mít testování průběh. Tedy, že postupně projdou 9 bludišti, kdy každé ovládní bude mít právě 3 bludiště s postupně se zvyšující obtížností (velikost, délka správné cesty, počet slepých konců). Dále, že je začátek bludiště označen zelenou průhlednou stěnou, jíž když projdou, tak se na pozadí spustí časomíra. Tato časomíra se pak zastaví a zobrazí se jim před očima při průchodu cílem, který je označen červenou průhlednou stěnou. Obeznámí se s tím, že musí čas průchodu bludištěm nadiktovat měřiteli, který ho zaznamená. Probandům budou také přehrány videa, která ukazují průchod demonstrativním bludištěm s použitím ovladačů a gest rukou, včetně startu, cíle a zobrazení času.

8.1.2 Testovací část

Testování ovládní bude probíhat v náhodném pořadí. Při přestupech mezi jednotlivými ovládními si proband bude moct udělat až pěti minutovou pauzu, která však nebude povinná. Tato pauza je zařazena z důvodu potenciálního výskytu cybersickness.

1) Průchod bludištěm pomocí Cybershoes

Před samotným používáním Cybershoes bude probandům vysvětlen princip tohoto ovládní. Proband usedne na otočnou židli umístěnou na koberci. Poté mu bude vysvětlen systém utahování a povolování Cybershoes (přezkové vázání). Nejdříve si nasadí návleky na boty Cybershoes a utáhne si je a následně si nastaví židli na správnou výšku tak, aby spodek Cybershoes lehce dotýkal koberce. Poté dostane Valve Index headset a ocitne se v základním prostoru SteamVR. V tuto chvíli se probandovi zapne tréninkové prostředí, ve kterém si

proband může vyzkoušet pohyb pomocí Cybershoes a ověří se správnost upnutí Cybershoes, tak aby testování nebylo ovlivněno jejich vyzutím. Po 2 minutách v tréninkovém prostředí bude probandovi zapnuto první bludiště pro Cybershoes, kterým projde a nadiktuje výsledný čas. Tento proces se zopakuje pro bludiště II. a III. kategorie s použitím Cybershoes.

2) Průchod bludištěm pomocí ovladačů

V tomto případě bude proband sedět na neotočné židli. Zabrání se tak otáčení bez využití ovladačů. Před samotným seznámením probanda s ovládáním při průchodu bludištěm se probandovi vysvětlí ovládání při navigaci systémem headsetu Oculus Quest 2. To z toho důvodu, že si proband bude sám zapínat a vypínat jednotlivé aplikace bludišť, které nalezne v knihovně aplikací pod názvy "Ovladace_I", "Ovladace_II", "Ovladace_III".

Poté mu bude vysvětleno, jak se ovládají ovladače při průchodu bludištěm, tj. pohyb pomocí joysticku na pravém ovladači a otáčení pomocí joysticku na levém ovladači o 45 stupňů. Proband postupně projde bludištěm od I. do III. kategorie s použitím ovladačů a opět po každém průchodu sdělí výsledný čas měřiteli.

3) Průchod bludištěm pomocí gest rukou

U tohoto ovládání proband stojí uprostřed volného prostoru, který musí mít minimální průměr 2 metry. Ve stručnosti se mu vysvětlí, co je hand-tracking, jak tato technologie funguje. Poté se seznámí s tím, jak se naviguje ve VR brýlích Oculus Quest 2 pomocí gest rukou, kde najde knihovnu aplikací a aplikace jednotlivých bludišť I, II a III pod názvy "Ruce_I", "Ruce_II", "Ruce_III". Také se probandovi ukáže, jak aplikaci vypnout po průchodu bludištěm a nadiktování času. Před samotným průchodem se probandovi znovu vysvětlí, jak funguje ovládání gesty rukou při průchodu bludišti. Tedy, že z pravé dlaně vychází paprsek, kterým míří na určité místo, a teleportace se provádí sevřením levé ruky v pěst. V poslední řadě se sdělí tipy na průchod bludištěm:

- Mířit vždy doprostřed chodby, tak aby se vyhnul kolizi se zdí.
- Při kolizi se zdí se stačí pouze vyklonit zpět do chodby a od teleportovat se do jejího středu, což se také může stihnout ještě před celkovým zatměním, jelikož je zde dvou vteřinová prodleva.

Proband postupně projde bludišti od I do III kategorie s použitím gest rukou a opět nadiktuje časy průchodu měřiteli.

8.1.3 Závěrečná část

Po dokončení všech 9 bludišť bude probandovi předložen dotazník ohledně průběhu testování. Do tohoto dotazníku pak uvedou své zkušenosti s VR, ohodnotí všechno ovládání na základě „System Usability Scale (SUS)“ a jejich vnímání cybersickness viz.

8.1.4 Testovací skupina

Testovací skupinu budou tvořit vysokoškolští studenti ve věku mezi dvaceti a dvaadvaceti lety, v celkovém počtu 15 lidí. Z něhož jednu třetinu budou tvořit ženy a dvě třetiny muži.

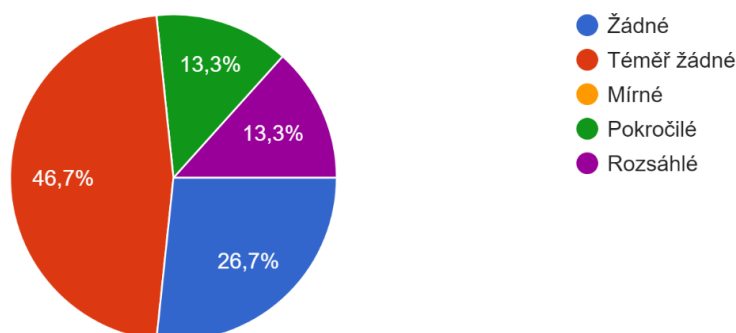
8.2 Průběh měření

Měření probíhalo podle plánované metodiky testování. Pro úsporu času při testech byli zahrnuti jednotlivci, ale také skupinky až o 3 lidech, kteří procházeli testem paralelně. Každému z nich bylo pečlivě a individuálně vysvětleno použití jednotlivých ovládaní v souladu s metodikou.

Během měření bylo pozorováno několik poznatků. Při použití ovladačů se lidé lépe pohybovali, když pevně drželi ruce s ovladači v jedné statické poloze a nepohybovali s nimi. Naopak lidé, kteří se volněji otáčeli a pohybovali ovladači v prostoru, měli větší obtíže se správným nasměrováním. Při ovládání pomocí rukou, se pak často objevoval problém „ztráty ve zdi“, kdy se probandi často teleportovali do zdi, kamera se zatměla a ztratili tak orientaci. I přes rychlé vrácení zpět na chodbu bludiště se stalo, že zvolili směr, ze kterého přišli a vraceli se zpět na začátek. To prodlužovalo čas průchodu. Při použití Cybershoes se neobjevovaly žádné výrazné problémy a subjektivně se toto ovládání jeví jako nejpřívětivější. Průměrná doba trvání testu pro jednotlivé účastníky byla předpokládána na 30 minut. Osoby s vysokými zkušenostmi s virtuální realitou byly schopny projít testem rychleji, z důvodu rychlé schopnosti navigovat se v systému brýlí Oculus Quest 2.

8.3 Výsledky měření

Prvním bodem průzkumu, byla zkušenost probandů s používáním virtuální reality.



Obrázek 56 Zkušenosti respondentů

Další bodem průzkumu bylo ohodnocení ovládaní v rámci orientace v prostoru, dle subjektivních zkušeností respondentů. Každý respondent pak rozdělil ovládaní od 1. po 3. místo. Ovládaní umístěné jako první dostalo 3 body, druhé 2 body a třetí 1 bod.

Tabulka 7 Pořadí dle orientace

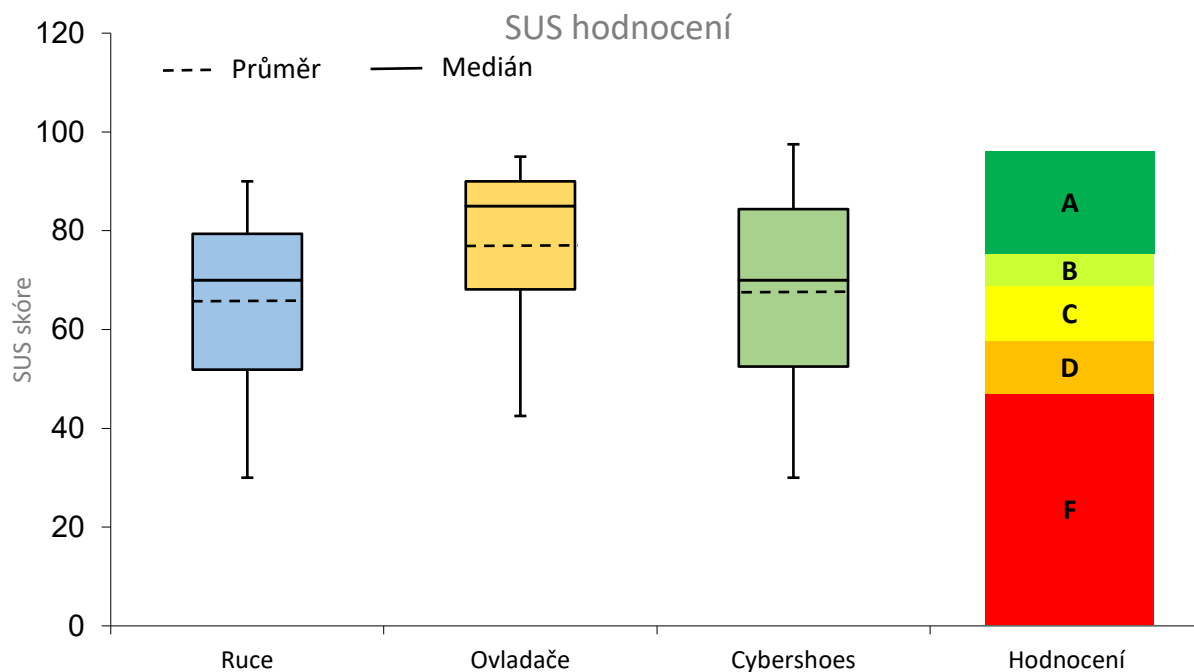
	1.	2.	3.	Body	Pořadí
Ruce	2	3	10	22	3.
Ovladače	7	6	2	35	1.
Cybershoes	6	6	3	33	2.

Dalším bodem dotazníku bylo SUS skóre jednotlivých ovládaní. Zde se bylo nejlépe hodnoceno ovládaní pomocí ovladačů se SUS skórem 74,67. Ovládaní Cybershoes dostalo hodnocení 67,83 a ovládaní rukama 65,83. Průměrné SUS skóre je 68. Z toho vyplývá, že pouze ovládaní ovladači se umístilo nad průměrným hodnocením systémů dle SUS.

Veškeré hodnoty z Tabulka 8 jsou znázorněny v (Obrázek 57).

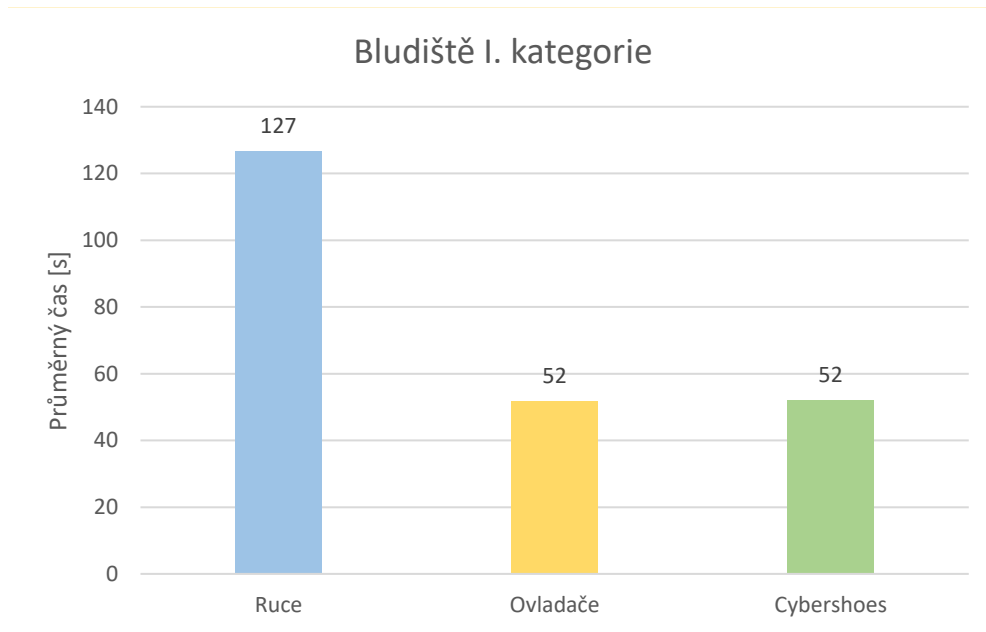
Tabulka 8 SUS hodnocení

Ovládání	SUS skóre (průměr)	Min	Max	Q ₁	Medián	Q ₃	Známka
Cybershoes	67,83	30	97,5	52,5	70	85	C
Ovladače	74,67	42,5	95	65	85	90	B
Ruce	65,83	30	90	50	70	80	C

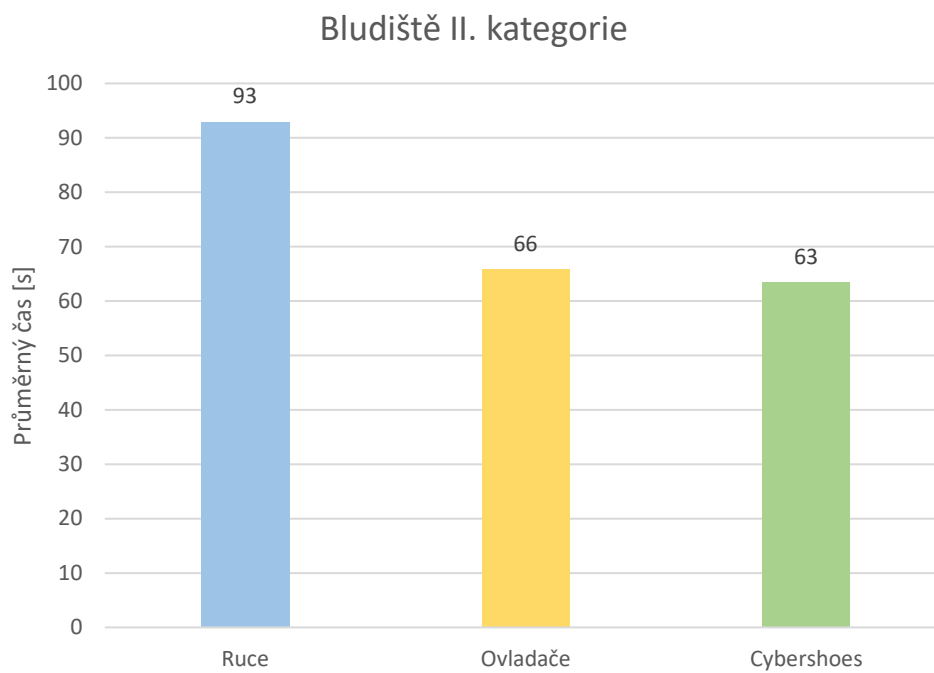


Obrázek 57 SUS hodnocení

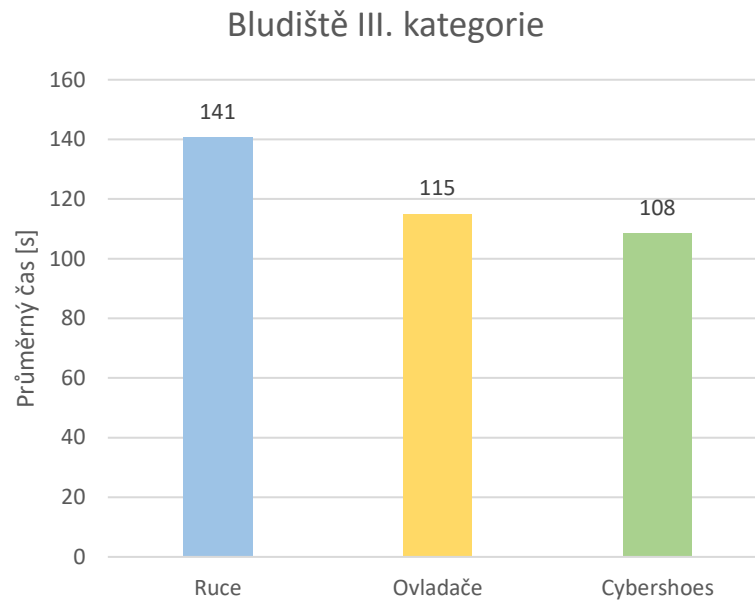
Výstupem měření jsou časy průchodu bludištěm. Následující grafy zobrazují rozdíly v průměrných časech průchodu bludišti pro jednotlivé kategorie pomocí různých ovládaní.



Obrázek 58 Průměrný čas I. Kategorie



Obrázek 59 Průměrný čas II. kategorie



Obrázek 60 Průměrný čas III. Kategorie

Následující tabulky (Tabulka 9, Tabulka 10, Tabulka 11) porovnávají rozdíly průměrných časů mezi jednotlivými ovládaními, tedy poukazují na korelace mezi jednotlivými variantami ovládaní. Veškeré časy, jsou uvedeny v sekundách. Data tabulky jsou rozdělená na tři kategorie bludišť I, II, III. První dva sloupce pod jednotlivými kategoriemi obsahují data průchodů bludišť od jednotlivých probandů. Tato data jsou seřazena dle hlavičky tabulky. Například pro Tabulka 9 jsou v prvních sloupcích každé kategorie průměrné časy od ovládaní rukama a v druhém pak průměrném časy ovládaní ovladači. Třetí sloupec pak udává procentuální rozdíl v časech mezi jednotlivými ovládaními. Kladné hodnoty jsou přiřazeny rychlejšímu ovládaní.

Tabulka 9 Ruce – Ovladače

Měření	Ruce – Ovladače								
	I			II			III		
1	162	40	305%	72	102	-42%	105	70	50%
2	103	132	-28%	133	151	-14%	160	368	-130%
3	339	69	391%	192	62	210%	347	42	726%
4	168	68	147%	56	55	2%	127	87	46%
5	68	28	143%	49	48	2%	127	79	61%
6	72	41	76%	64	54	19%	79	87	-10%
7	169	53	219%	55	54	2%	89	106	-19%
8	74	54	37%	172	78	121%	146	256	-75%
9	42	39	8%	158	57	177%	214	100	114%
10	64	39	64%	58	51	14%	151	59	156%
11	64	40	60%	80	54	48%	98	75	31%
12	89	56	59%	115	40	188%	172	210	-22%
13	69	50	38%	49	40	23%	80	59	36%
14	213	33	545%	63	91	-44%	102	39	162%
15	203	32	534%	78	51	53%	111	88	26%
Průměrný čas	127	52	144%	93	66	41%	141	115	23%
Průměrný rozdíl	75			27			26		
Rychlejší ovládání	Ovladače			Ovladače			Ovladače		

Tabulka 10 Ruce – Cybershoes

Měření	Ruce – Cybershoes								
	I			II			III		
1	162	52	212%	72	54	33%	105	110	-5%
2	103	133	-29%	133	157	-18%	160	130	23%
3	339	77	340%	192	122	57%	347	197	76%
4	168	39	331%	56	50	12%	127	84	51%
5	68	71	-4%	49	44	11%	127	67	90%
6	72	41	76%	64	50	28%	79	91	-15%
7	169	38	345%	55	57	-4%	89	159	-79%
8	74	33	124%	172	46	274%	146	66	121%
9	42	33	27%	158	39	305%	214	67	219%
10	64	60	7%	58	67	-16%	151	160	-6%
11	64	54	19%	80	68	18%	98	84	17%
12	89	26	242%	115	46	150%	172	69	149%
13	69	21	229%	49	35	40%	80	64	25%
14	213	38	461%	63	56	13%	102	127	-25%
15	203	62	227%	78	60	30%	111	151	-36%
Průměrný čas	127	52	144%	93	63	48%	141	108	31%
Průměrný rozdíl	75			30			33		
Rychlejší ovládání	Cybershoes			Cybershoes			Cybershoes		

Tabulka 11 Ovladače – Cybershoes

Měření	Ovladače – Cybershoes								
	I			II			III		
1	40	52	-30%	102	54	89%	70	110	-57%
2	132	133	-1%	151	157	-4%	368	130	183%
3	69	77	-12%	62	122	-97%	42	197	-369%
4	68	39	74%	55	50	10%	87	84	4%
5	28	71	-154%	48	44	9%	79	67	18%
6	41	41	0%	54	50	8%	87	91	-5%
7	53	38	39%	54	57	-6%	106	159	-50%
8	54	33	64%	78	46	70%	256	66	288%
9	39	33	18%	57	39	46%	100	67	49%
10	39	60	-54%	51	67	-31%	59	160	-171%
11	40	54	-35%	54	68	-26%	75	84	-12%
12	56	26	115%	40	46	-15%	210	69	204%
13	50	21	138%	40	35	14%	59	64	-8%
14	33	38	-15%	91	56	63%	39	127	-226%
15	32	62	-94%	51	60	-18%	88	151	-72%
Průměrný čas	52	52	0%	66	63	5%	115	108	6%
Průměrný rozdíl	0		0%	3		5%	7		6%
Rychlejší ovládání	Stejně			Cybershoes			Cybershoes		

Poslední část dotazníku se zaměřovala na vnímání cybersickness respondentů. Každý respondent byl dotázán na to, zda při použití konkrétního ovládání pocítil cybersickness. Pokud ano, byl požádán, aby na škále od 1 do 5 vyjádřil míru této nevolnosti. Tabulka 12 pak obsahuje škálové hodnocení, které zároveň slouží jako bodové hodnocení. Pokud respondent cybersickness nepocítil, pole je proškrtnuté.

Tabulka 12 Cybersickness hodnocení

Měření	Ruce	Ovladače	Cybershoes
1	–	3	5
2	4	5	4
3	5	3	5
4	–	–	2
5	–	–	–
6	–	2	–
7	–	4	4
8	2	5	1
9	4	4	3
10	–	–	3
11	3	–	3
12	–	–	–
13	–	–	–
14	–	–	–
15	–	–	3
Ano	5x	7x	10x
Body	23	26	33
Pořadí	1.	2.	3.

Ověření nulové hypotézy

Hypotéza byla stanovena v Úvodu (1) práce. Ověření se provede dle Kruskal-Wallisova testu.

Vstupní data, průměrné časy vzaté ze všech kategorií jednotlivých ovládnání:

Tabulka 13 Průměrné časy ze 3 kategorií

Ruce	Ovladače	Cybershoes
113	71	72
132	217	140
293	58	132
117	70	58
81	52	61
72	61	61
104	71	85
131	129	48
138	65	46
91	50	96
81	56	69
125	102	47
66	50	40
126	54	74
131	57	91

Prvním krokem testu je seřazení veškerých hodnot nezávisle na jejich skupině a nahradit je příslušnými pořadími. Současně spočítat součet jednotlivých pořadí. Pro zjednodušení výpočtu byla také určena hodnota počtu pozorování a následně kvadrát součtu jednotlivých pořadí vydělen počtem pozorování. Tyto hodnoty se uvádí pro každý sloupec zvlášť viz Tabulka 14.

Tabulka 14 Pořadí

		Pořadí		
		Ruce	Ovladače	Cybershoes
		33	21	23
		41	44	43
		45	12	41
		34	19	12
		26	7	14
		23	14	14
		32	21	27
		39	37	4
		42	16	2
		29	6	30
		26	9	18
		35	31	3
		17	6	1
		36	8	24
		39	10	29
Součet pořadí	R	494	259	283
Počet pozorování v i-té třídě	n	15	15	15
	R ² /n	16236,2	4454,8	5339,3

Posledními potřebnými hodnotami pro výpočet je počet pozorování ve všech třídách $N = 45$ a počet tříd $k = 3$.

Posledním krokem je určit testovací kritérium H pomocí vzorce:

$$H = \frac{12}{N(N + 1)} \sum_{i=1}^k \frac{R_i^2}{n_i} - 3(N + 1) = 12,9$$

Pro tuto hodnotu H pak z tabulek kritických hodnot χ^2 najdeme kritickou hranici:

$$\chi_{\alpha 3-1}^2 = 5,99147$$

Protože hodnota testovací statistiky H je vyšší než kritická hodnota, zamítáme nulovou hypotézu. To znamená, že existují statisticky významné rozdíly v průměrných časech průchodů bludišť mezi všemi třemi ovládaními.

8.4 Vyhodnocení

Z (Obrázek 56) je pak zřejmé, že téměř polovina respondentů již minimální zkušenosti s VR měla. Zhruba čtvrtina pak zkušenosti neměla žádné a další čtvrtina má s VR nadprůměrné zkušenosti. Zde je důležité podotknout, že rozdíl v časech průchodu se u lidí s většími zkušenostmi nijak výrazně nelišil od lidí se zkušenostmi žádnými či téměř žádnými. To může být dobrým ukazatelem celkové intuitivnosti každého z ovládání.

Z výsledků v Tabulka 7, nejlépe vyšly ovladače s celkovým počtem 35 bodů. Cybershoes získali o 2 body méně. Ovládání rukama, získalo o 13 bodů méně než ovladače. Z těchto výsledků vyplývá, že co se orientace týče, jsou ovladače a Cybershoes téměř srovnatelné. Z pozorování však vyšel jeden důležitý fakt, že při pohybu pomocí Cybershoes se probandi rozhlíželi v bludišti volněji a snadněji než při využití ostatních ovládání. To je způsobeno tím, že použití Cybershoes nevyžaduje soustředění na ovládací prvky, protože chůze s nimi je velmi podobná běžné chůzi, na kterou jsou lidé zvyklí. U ovládání ovladači pak byli nuceni zastavovat a rozhlížet se z klidu. Tento jev se také promítl do celkových časů, jak bude dále zmíněno. Ovládání rukama získalo o 13 bodů méně než ovladače. Tento rozdíl je především způsoben rychlými změnami pohybu při použití teleportace. Zatímco u prvních dvou ovladačů byl pohyb kontinuální, při teleportaci byl pohyb skokový, což vedlo k časté ztrátě orientace.

Značnou částí dotazníků bylo hodnocení pomocí SUS skóre. Z tohoto hodnocení vyplývá, že jediným systémem, který dosáhl nadprůměrného skóre dle průměru SUS, jsou ovladače. Cybershoes jsou o 0,17 bodu pod průměrem a ovládání rukama dokonce o 2,17 bodu pod průměrem. U ovládání rukama by mělo dojít k vylepšení systému ovládání. Zejména by bylo vhodné upravit zatemňování, které způsobovalo u mnoha účastníků ztrátu orientace. Je třeba také poznamenat, že tento faktor je poněkud zavádějící, protože nemá přímý vliv na samotné ovládání a mohl tak záporně ovlivňovat jeho hodnocení. Další možnou úpravou by mohla být změna křivky paprsku znázorňujícího laser, která by mohla být více lineární. Tato změna by mohla zlepšit odhad vzdálenosti teleportace a probandi by se tak vyhnuli častým kolizím se zdmi. Celkové výsledky hodnocení pomocí SUS jsou prezentovány v Tabulka 8 a vizualizovány na (Obrázek 57).

Po průchodu bludišti vznikl výstup v podobě času. Při porovnání průměrných časů průchodu u jednotlivých kategorií bludišť, jak je znázorněno v (Obrázek 58 - Obrázek 60) se ukázalo, že ovládání pomocí rukou bylo nejpomalejší napříč všemi kategoriemi. Naopak ovládání pomocí Cybershoes dosahovalo nejrychlejších průměrných časů. Z Tabulka 9 je zřejmé, že ovládání pomocí ovladačů bylo rychlejší než ovládání pomocí rukou ve všech kategoriích. Rozdíl v průměrném čase byl v první kategorii 75 sekund, ve druhé kategorii 27 sekund a ve třetí kategorii 26 sekund. Z Tabulka 10 je patrné, že ovládání pomocí Cybershoes bylo také rychlejší než ovládání pomocí rukou ve všech kategoriích s ještě většími rozdíly. Průchodové časy pomocí ovladačů a Cybershoes se velmi podobaly. V první kategorii byly průměrné časy totožné. Ve druhé a třetí kategorii bylo ovládání pomocí Cybershoes o 2 a 7 sekund rychlejší viz Tabulka 11. Z procentuálního rozdílu napříč tabulkami můžeme sledovat zmenšující se trend v rozdílu jednotlivých časů s narůstající náročností bludiště. Zároveň byla otestována a zamítnuta nulová hypotéza, která se stanovila na začátku práce.

Posledním výzkumným bodem bylo porovnání výskytu virtuální nevolnosti při použití jednotlivých ovládání. Z Tabulka 12 je patrné, že při ovládání pomocí rukou ji zaznamenala jedna třetina respondentů. Při použití ovladačů se virtuální nevolnost projevila u 7 z 10 respondentů a při použití Cybershoes u 10 z 15 respondentů. Největší výskyt virtuální nevolnosti byl zaznamenán při použití Cybershoes. To potvrzuje pozorování z testování, kdy 75 % žen mělo s tímto ovládáním výrazné potíže a trpěly velkou mírou virtuální nevolnosti.

9 Závěr

Cílem této bakalářské práce bylo porovnat tři různé druhy ovládání ve virtuálním prostoru z hlediska orientace. Práce se skládá z několika částí.

V první části byla provedena teoretická rešerše, která se zaměřila na problematiku orientace ve virtuální realitě. Dále byla popsána jednotlivá testovaná ovládání a jejich vlastnosti. Posledním aspektem byla prezentace bludišť, která byla využita jako simulační prostředí pro testování.

Druhá část práce byla praktická. Pro účely testování bylo nutné vytvořit model simulačního prostředí v herním enginu Unity. Modely jednotlivých prvků bludišť byly vytvořeny v software Blender a následně integrovány do Unity. Funkcionalita prostředí, včetně časomíry a startovní pozice, byla implementována pomocí programovacího jazyka C# ve vývojovém prostředí Visual Studio.

Třetí částí práce zahrnuje testování. Nejprve byla sestavena metodika testování, podle které bylo testování prováděno na vzorku 15 osob. Měřily se časy průchodu jednotlivých bludišť a pomocí dotazníku byla hodnocena orientace respondentů, vnímání cybersickness a uživatelská přívětivost ovládání pomocí SUS skóre. Z dat naměřených časů se otestovala a následně zamítla nulová hypotéza stanovená na začátku práce. Mezi hodnotami průměrných časů ze všech třech kategorií u ovládání Ruce, Ovladače a Cybershoes tedy existují statisticky významné rozdíly.

Poslední část práce se zaměřila na celkové vyhodnocení výsledků.

Bakalářská práce poskytuje vhodný vstupní bod pro další testování, které by mohlo zahrnovat rozšíření vzorku testovaných osob. Důležité je zaměřit se na rozdíly ve věku, pohlaví a dalších relevantních faktorech, které mohou ovlivnit vnímání a schopnost orientace ve virtuálním prostoru. Dalším krokem by mělo být rozšíření spektra testovaných ovládání. Například by se mohl zahrnout systém teleportace pro ovladače nebo systém kontinuálního pohybu založený na gestech rukou. Dále je také nutné upravit systém zatemňování při teleportaci, který byl identifikován jako faktor ovlivňující výsledky testování. Zde by mohlo pomoci neúplné zatmění, ale jen částečné, tak aby se uživatel mohl lépe vrátit zpět na chodbu bludiště.

10 Seznam použité literatury

- [1] GOLLEDGE, R.G. *Wayfinding Behavior: Cognitive Mapping and Other Spatial Processes: Wayfinding Behavior*. Johns Hopkins University Press, 1999. ISBN 9781421402895. Dostupné také z: <https://books.google.cz/books?id=FRFcQZYij4QC>
- [2] PSTEL, S., C. CHEN, D. BÜRGER, M. NAUJOKS, L. MARTIN, K. PETRI a K. WITTE. Spatial orientation in virtual environment compared to real-world. *Journal of Motor Behavior* [online]. 2021, **53**(6), 693-706 [cit. 2022-11-27]. ISSN 0022-2895. Dostupné z: [doi:10.1080/00222895.2020.1843390](https://doi.org/10.1080/00222895.2020.1843390)
- [3] VR Headset Field of View Comparison. In: *Virtualrealityobserver.com* [online]. Las Vegas: GenVR, Inc., 2022 [cit. 2022-11-27]. Dostupné z: <https://virtualrealityobserver.com/vr-headset-field-of-view-comparison/>
- [4] REBENITSCH, Lisa. Managing cybersickness in virtual reality. *XRDS: Crossroads, The ACM Magazine for Students* [online]. 2015, **22**(1), 46-51 [cit. 2022-11-29]. ISSN 1528-4972. Dostupné z: [doi:10.1145/2810054](https://doi.org/10.1145/2810054)
- [5] CHATTHA, Umer, Uzair JANJUA, Fozia ANWAR, Tahir MADNI, Muhammad CHEEMA a Sana JANJUA. Motion Sickness in Virtual Reality: An Empirical Evaluation. *IEEE Access* [online]. 2020, **8**, 130486-130499 [cit. 2022-11-29]. ISSN 2169-3536. Dostupné z: [doi:10.1109/ACCESS.2020.3007076](https://doi.org/10.1109/ACCESS.2020.3007076)
- [6] LECON, Carsten. Motion Sickness in VR Learning Environments [online]. 2019/08/22. Athens: Athens Institute for Education and Research, 2018/08/30, 13 s. [cit. 2023-05-25]. ATINER'S Conference Paper Series. ISSN 2241-2891. 2241-2891. Dostupné z: <http://www.atiner.gr/papers/COM2018-2514.pdf>
- [7] BUHLER, Helmut, Sebastian MISZTAL a Jonas SCHILD. Reducing VR Sickness Through Peripheral Visual Effects. In: *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* [online]. IEEE, 2018, s. 517-9 [cit. 2022-11-29]. ISBN 978-1-5386-3365-6. Dostupné z: [doi:10.1109/VR.2018.8446346](https://doi.org/10.1109/VR.2018.8446346)
- [8] GRAEBER, David a Kay STANNEY. Gender Differences in Visually Induced Motion Sickness. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* [online]. 2002, **46**(26), 2109-2113 [cit. 2022-11-29]. ISSN 2169-5067. Dostupné z: [doi:10.1177/154193120204602602](https://doi.org/10.1177/154193120204602602)
- [9] ARNS, L.L. a M.M. CERNEY. The relationship between age and incidence of cybersickness among immersive environment users. In: *IEEE Proceedings. VR 2005. Virtual Reality, 2005* [online]. IEEE, 2005, s. 267-268 [cit. 2022-11-29]. ISBN 0-7803-8929-8. Dostupné z: [doi:10.1109/VR.2005.1492788](https://doi.org/10.1109/VR.2005.1492788)
- [10] *Meta* [online]. Meta, 2022 [cit. 2022-12-02]. Dostupné z: <https://about.meta.com/>
- [11] Hand Tracking. In: *Developer.oculus* [online]. Meta, 2022 [cit. 2022-12-01]. Dostupné z: <https://developer.oculus.com/resources/hands-design-intro/>
- [12] Quest Networked Hand Tracking. In: *Uploadvr* [online]. Ian Hamilton, 2020 [cit. 2022-12-01]. Dostupné z: <https://uploadvr.com/network-hand-tracking-quest/>

- [13] Input Modalities. In: *Developer.oculus* [online]. Meta, 2022 [cit. 2022-12-01]. Dostupné z: <https://developer.oculus.com/documentation/unity/unity-handtracking/>
- [14] Best Practices. In: *Developer.oculus* [online]. Meta, 2022 [cit. 2022-12-02]. Dostupné z: <https://developer.oculus.com/resources/hands-design-bp/>
- [15] Hand States and Gates. In: *Developer.oculus* [online]. Meta, 2022 [cit. 2022-12-01]. Dostupné z: <https://developer.oculus.com/resources/hands-design-bp/>
- [16] Tracking system. In: *Mechatech* [online]. Stockport: Mechatech, 2021 [cit. 2022-12-02]. Dostupné z: <https://www.mechatech.co.uk/journal/how-do-common-virtual-reality-tracking-systems-work>
- [17] *Circuit Stream* [online]. CALGARY: Circuit Stream, 2022 [cit. 2022-12-02]. Dostupné z: <https://www.circuitstream.com/>
- [18] Location of features on controllers. In: *Vrteamspace* [online]. Crowthorne: Box Bear Digital Ltd, 2020 [cit. 2022-12-02]. Dostupné z: <https://www.vrteamspace.co.uk/setup/>
- [19] BROWN, Rayen. Cybershoes. In: *VRcompare* [online]. VRcompare, 2021 [cit. 2022-12-02]. Dostupné z: <https://vr-compare.com/accessory/cybershoes>
- [20] Cybershoes. In: *Steampowered* [online]. USA: Valve Corporation, 2022 [cit. 2022-12-02]. Dostupné z: <https://store.steampowered.com/app/1222360/Cybershoes/?l=czech>
- [21] Jaký je rozdíl mezi Labyrintem a Bludištěm?. In: *Místo v srdci* [online]. Česká Republika: Místo v srdci, 2020 [cit. 2022-12-02]. Dostupné z: <https://mistovsrdci.cz/terapie-tmou-pobyt-ve-tme/jaky-je-rozdil-mezi-labyrintem-a-bludistem/>
- [22] MATĚJKA, Petr. *Algoritmy pro generování a řešení bludišť*. Brno, 2012. Diplomová práce. Masarykova univerzita, Fakulta informatiky. Vedoucí práce Radek PELÁNEK.
- [23] D. PULLEN, Walter. Maze Classification. In: *Astrolog* [online]. astrolog, 2022 [cit. 2022-12-02]. Dostupné z: <http://www.astrolog.org/labyrnth/algorithm.htm>
- [24] *DoYouMaze?* [online]. USA: doyoumaze, 2020 [cit. 2022-12-02]. Dostupné z: <https://www.doyoumaze.com/>
- [25] *Unity* [online]. Unity Technologies, 2022 [cit. 2022-12-02]. Dostupné z: <https://unity.com/>
- [26] UMBRA EVOLUTION. Umbra Maze Magician. In: *Unity Asset Store* [online]. Unity Technologies, 2023 [cit. 2023-03-26]. Dostupné z: <https://assetstore.unity.com/packages/tools/level-design/umbra-maze-magician-39540#description>
- [27] BATSOFT. Core Maze. In: *Unity Asset Store* [online]. Unity Technologies, 2023 [cit. 2023-03-26]. Dostupné z: <https://assetstore.unity.com/packages/tools/utilities/core-maze-225869#description>
- [28] JESTERWAY SOFTWARE. Custom Maze Generator. In: *Unity Asset Store* [online]. Unity Technologies, 2023 [cit. 2023-03-26]. Dostupné z: <https://assetstore.unity.com/packages/tools/level-design/custom-maze-generator-209172#description>

- [29] REYNOLDS, Max. Solving Mazes With Python: Using Dijkstra's Algorithm and OpenCV. In: *Medium* [online]. United Kingdom: VeraSafe United Kingdom Ltd., 2023 [cit. 2023-03-27]. Dostupné z: <https://medium.com/towards-data-science/solving-mazes-with-python-f7a412f2493f>
- [30] HARABOR, Daniel a Alban GRASTIEN. Online Graph Pruning for Pathfinding On Grid Maps. *Proceedings of the AAAI Conference on Artificial Intelligence* [online]. 2011, 25(1), 1114-1119 [cit. 2023-03-27]. ISSN 2374-3468. Dostupné z: doi:10.1609/aaai.v25i1.7994
- [31] MOORE, Terrence a Jin-Hee CHO. Applying Percolation Theory. In: KOTT, Alexander a Igor LINKOV, ed., Alexander KOTT, Igor LINKOV. *Cyber Resilience of Systems and Networks* [online]. Cham: Springer International Publishing, 2019, s. 107-133 [cit. 2023-03-27]. ISBN 978-3-319-77491-6. Dostupné z: doi:10.1007/978-3-319-77492-3_6
- [32] SteamVR_CyberRunner_Unity. In: *GitHub* [online]. San Francisco, California, U.S.: Microsoft, 2008 [cit. 2023-05-18]. Dostupné z: https://github.com/CybershoesVR/SteamVR_CyberRunner_Unity/tree/master/Assets/Project/Scripts/Player
- [33] HROMÁDKOVÁ, Lada. *Šilhání*. Vyd. 2., dopl. Brno: Institut pro další vzdělávání pracovníků ve zdravotnictví, 1995. ISBN 80-701-3207-8.
- [34] HRUŠKOVÁ, Jana, Juraj JAKUBÍK, Michal HENDRYCH, Richard PAŠTĚKA, Jana SVAČINOVÁ, Ksenia BUDINSKAYA, Veronika KUJALOVÁ, Veronika VEJTASOVÁ a Zuzana NOVÁKOVÁ. *Fyziologie – teorie k praktickým cvičením* [online]. 1. vyd. Brno: Masarykova univerzita, 2021 [cit. 2022-11-27]. Elportál. Dostupné z: <http://is.muni.cz/elportal/?id=1783338>. ISSN 1802-128X
- [35] DVORSKÝ, Lukáš. *Vliv pseudo 3D a 3D zobrazení na vnímání hloubky* [online]. Brno, 2015 [cit. 2022-11-27]. Dostupné z: <https://is.muni.cz/th/bziyi/>. Diplomová práce. Masarykova univerzita, Filozofická fakulta. Vedoucí práce Čeněk ŠAŠINKA
- [36] Kinetóza. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2022-11-29]. Dostupné z: <https://cs.wikipedia.org/w/index.php?title=Kinet%C3%B3za&oldid=20335773>
- [37] Triangulace. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2022 [cit. 2022-12-02]. Dostupné z: <https://cs.wikipedia.org/w/index.php?title=Triangulace&oldid=20082660>

PŘÍLOHA č. 1

Dotazník

Jaké jsou Vaše zkušenosti s používáním virtuální reality? *

Žádné - Nikdy jsem virtuální realitu nevyužil/a.

Téměř žádné - Ve virtuální realitě jsem mohl/a strávit do 5 hodin.

Mírné - Ve virtuální realitě jsem mohl/a strávit více jak 5 hodin.

Pokročilé - Ve virtuální realitě jsem mohl/a strávit více jak 20 hodin.

Rozsáhlé - Ve virtuální realitě jsem mohl/a strávit více jak 60 hodin.

- Žádné
- Téměř žádné
- Mírné
- Pokročilé
- Rozsáhlé

Jak byste seřadili jednotlivá ovládání dle orientace v prostoru? *

	Ruce	Ovladače	Cybershoes
1.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Ovládání za pomoci gest rukou

Myslím, že bych toto ovládání rád/a používal/a často. *

	1	2	3	4	5	
Naprostou nesouhlasím	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Naprostou souhlasím

Ovládání mi připadalo zbytečně složité. *

	1	2	3	4	5	
Naprosto nesouhlasím	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Naprosto souhlasím

Ovládání se mi zdálo snadno použitelné. *

	1	2	3	4	5	
Naprosto nesouhlasím	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Naprosto souhlasím

Myslím, že bych potřeboval/a podporu technické osoby, abych mohl/a toto ovládání používat. *

	1	2	3	4	5	
Naprosto nesouhlasím	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Naprosto souhlasím

Jednotlivé funkce tohoto ovládání mi připadaly dobře integrované. *

	1	2	3	4	5	
Naprosto nesouhlasím	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Naprosto souhlasím

Myslím, že při tomto ovládání bylo příliš mnoho nesrovnalostí. *

	1	2	3	4	5	
Naprosto nesouhlasím	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Naprosto souhlasím

Dovedu si představit, že většina lidí by se toto ovládání naučila používat velmi rychle. *

	1	2	3	4	5	
Naprosto nesouhlasím	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Naprosto souhlasím

Používání ovládání mi připadalo velmi krkolonné. *

	1	2	3	4	5	
Naprostο nesouhlasím	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Naprostο souhlasím

Při používání ovládání jsem se cítil/a velmi jistě. *

	1	2	3	4	5	
Naprostο nesouhlasím	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Naprostο souhlasím

Musel/a jsem se naučit spoustu věcí, než jsem mohl začít s tímto ovládáním pracovat. *

	1	2	3	4	5	
Naprostο nesouhlasím	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Naprostο souhlasím

Ovládání pomocí ovladačů

Myslím, že bych toto ovládání rád/a používal/a často. *

	1	2	3	4	5	
Naprostο nesouhlasím	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Naprostο souhlasím

Ovládání mi připadalo zbytečně složité. *

	1	2	3	4	5	
Naprostο nesouhlasím	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Naprostο souhlasím

Ovládání se mi zdálo snadno použitelné. *

	1	2	3	4	5	
Naprostο nesouhlasím	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Naprostο souhlasím

Myslím, že bych potřeboval/a podporu technické osoby, abych mohl/a toto ovládání používat. *

	1	2	3	4	5	
Naprostο nesouhlasím	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Naprostο souhlasím

Jednotlivé funkce tohoto ovládání mi připadaly dobře integrované. *

	1	2	3	4	5	
Naprostο nesouhlasím	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Naprostο souhlasím

Myslím, že při tomto ovládání bylo příliš mnoho nesrovnalostí. *

	1	2	3	4	5	
Naprostο nesouhlasím	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Naprostο souhlasím

Dovedu si představit, že většina lidí by se toto ovládání naučila používat velmi rychle. *

	1	2	3	4	5	
Naprostο nesouhlasím	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Naprostο souhlasím

Používání ovládání mi připadalo velmi krkolonné. *

	1	2	3	4	5	
Naprostο nesouhlasím	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Naprostο souhlasím

Při používání ovládání jsem se cítil/a velmi jistě. *

	1	2	3	4	5	
Naprostο nesouhlasím	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Naprostο souhlasím

Musel/a jsem se naučit spoustu věcí, než jsem mohl začít s tímto ovládním pracovat. *

	1	2	3	4	5	
Naprosto nesouhlasím	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Naprosto souhlasím

Ovládání využitím cybershoes

Myslím, že bych toto ovládání rád/a používal/a často. *

	1	2	3	4	5	
Naprosto nesouhlasím	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Naprosto souhlasím

Ovládání mi připadalo zbytečně složité. *

	1	2	3	4	5	
Naprosto nesouhlasím	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Naprosto souhlasím

Ovládání se mi zdálo snadno použitelné. *

	1	2	3	4	5	
Naprosto nesouhlasím	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Naprosto souhlasím

Myslím, že bych potřeboval/a podporu technické osoby, abych mohl/a toto ovládání používat. *

	1	2	3	4	5	
Naprosto nesouhlasím	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Naprosto souhlasím

Jednotlivé funkce tohoto ovládání mi připadaly dobře integrované. *

	1	2	3	4	5	
Naprosto nesouhlasím	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Naprosto souhlasím

Myslím, že při tomto ovládnání bylo příliš mnoho nesrovnalostí. *

	1	2	3	4	5	
Naprostο nesouhlasím	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Naprostο souhlasím

Dovedu si představit, že většina lidí by se toto ovládnání naučila používat velmi rychle. *

	1	2	3	4	5	
Naprostο nesouhlasím	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Naprostο souhlasím

Používání ovládnání mi připadalo velmi krkolomné. *

	1	2	3	4	5	
Naprostο nesouhlasím	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Naprostο souhlasím

Při používání ovládnání jsem se cítil/a velmi jistě. *

	1	2	3	4	5	
Naprostο nesouhlasím	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Naprostο souhlasím

Musel/a jsem se naučit spoustu věcí, než jsem mohl začít s tímto ovládnáním pracovat. *

	1	2	3	4	5	
Naprostο nesouhlasím	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Naprostο souhlasím

Zaznamenali jste cybersickness při ovládnání rukami? *

*Stav nevolnosti, závratě a podobných příznaků, které se mohou vyskytnout při používání virtuální reality (VR).

Ano

Ne

V jaké míře? *

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Zaznamenali jste cybersickness při ovládání ovladači? *

*Stav nevolnosti, závratě a podobných příznaků, které se mohou vyskytnout při používání virtuální reality (VR).

Ano

Ne

V jaké míře? *

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Zaznamenali jste cybersickness při ovládání pomocí cybershoes? *

*Stav nevolnosti, závratě a podobných příznaků, které se mohou vyskytnout při používání virtuální reality (VR).

Ano

Ne

V jaké míře? *

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>