



Zadání bakalářské práce

Název:	Virtuální logické hry
Student:	Lukáš Bárta
Vedoucí:	doc. Ing. Mgr. Petr Klán, CSc.
Studijní program:	Informatika
Obor / specializace:	Informační systémy a management
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	do konce letního semestru 2022/2023

Pokyny pro vypracování

Analýzujte desktopové logické hry a navrhnete metodiku pro jejich rozšíření do virtuálního prostředí. Na základě metodiky vytvořte koncept nové čistě logické hry, odvodte její payoff matici a hru ve virtuálním prostředí naprogramujte. Analyzujte optimální herní strategie. Postupujte podle následujících kroků:

1. Provedte analýzu desktopových logických her.
2. Navrhnete metodiku rozšíření logických her do virtuální reality.
3. Provedte průzkum trhu s cílem nalezení cílové skupiny uživatelů.
4. Na základě předchozí analýzy a průzkumu vytvořte koncept nové čistě virtuální logické hry.
5. Odvodte a analyzujte payoff matici hry z pohledu možných strategií.
6. Sestavte a naprogramujte novou hru ve virtuálním prostředí.
7. Otestujte novou logickou hru.
8. Naleznete optimální herní strategie nové hry.
9. Zhodnoťte navrženou metodiku a uveďte závěry, jak zlepšit desktopové logické hry.
10. Navrhnete finanční plán pro udržitelný rozvoj a šíření nové logické hry.

Bakalářská práce

VIRTUÁLNÍ LOGICKÉ HRY

Lukáš Bárta

Fakulta informačních technologií
Katedra softwarového inženýrství
Vedoucí: doc. Ing. Mgr. Petr Klán, CSc.
11. května 2022

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2022 Lukáš Bárta. Všechna práva vyhrazena..

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

Odkaz na tuto práci: Bárta Lukáš. *Virtuální logické hry*. Bakalářská práce. České vysoké učení technické v Praze, Fakulta informačních technologií, 2022.

Obsah

Seznam obrázků	v
Seznam tabulek	v
Poděkování	vi
Prohlášení	vii
Abstrakt	viii
Seznam zkratk	ix
Úvod	1
1 Cíl práce	3
2 Virtuální realita a logické hry	5
2.1 Virtuální realita	5
2.1.1 Historie	5
2.1.2 Využití VR	6
2.2 Logické hry	6
2.2.1 Kerbal Space Program	7
2.2.2 Besiege	9
2.2.3 Logické hry ve VR	10
3 Průzkum trhu	11
3.1 Celkový trh	11
3.2 Cílový trh	12
3.3 Sekundární průzkum	12
4 Použité technologie	15
4.1 Unity	15
4.1.1 Inspektor	15
4.1.2 Prefab a Prefab Variant	15
4.1.3 Komponenty	15
4.1.4 Test Framework	16
4.2 OpenXR	16
4.3 Blender	16
5 Návrh, implementace a testování	17
5.1 Návrh	17
5.1.1 Metodika rozšíření do VR	17
5.1.2 Editor	18
5.1.3 Logický systém	18
5.2 Implementace	19

5.2.1	Díly a fyzikální systém	19
5.2.2	Logický systém	19
5.2.3	Beziérový křivky	21
5.2.4	Generování meshe propojovacího kabelu	22
5.2.5	Ovládání	25
5.2.6	Modelování	25
5.3	Testování	26
5.3.1	Automatizované testování	26
5.3.2	Manuální testování	26
5.3.3	Interní testování	27
6	Optimální herní strategie	29
6.1	Teorie her	29
6.2	Optimální herní strategie	30
7	Finanční plán	33
7.1	Zahajovací rozvaha	33
7.2	Odhad výdajů a zisků	34
7.2.1	Realistický odhad	34
7.2.2	Optimistický odhad	35
7.2.3	Pesimistický odhad	35
8	Diskuze a závěr	37
	Bibliografie	39
	Obsah přiloženého média	43

Seznam obrázků

2.1	KSP herní cyklus	7
2.2	Ukázka z editoru KSP	8
2.3	Besiege herní cyklus	9
2.4	Ukázka z editoru Besiege	9
3.1	Projekce a historická data prodeje VR HMD a příjmů z obsahu pro VR [5]	12
4.1	Ukázka testovacího nástroje	16
5.1	Ukázka pravdivostní tabulky obvodu pro násobení signálu minus jedničkou	20
5.2	Příklad neřešitelného stavu způsobeným cyklem	21
5.3	Ukázka chyby generování meshe způsobené rotací normálových vektorů	23
5.4	Příklad náhlých rotací normálových vektorů [1]	23
5.5	Příklad minimalizovaných rotací normálových vektorů [29]	24
5.6	Ukázka směru zápisu Trojúhelníků	24
5.7	Ukázka propojování kabelem	24
5.8	Ukázka mačkání tlačítka	25
5.9	Ukázka modelů ze hry	25
5.10	Ukázka testování obvodů – implementace NAND brány	27
5.11	Ukázka Gizmos pro brány	27
6.1	Ukázka vozítka se zatáčením vpředu	31
6.2	Ukázka vozítka se zatáčením vzadu	31

Seznam tabulek

3.1	Ceny populárních her	13
6.1	Payoff matice pro věžňovo dilema	30
6.2	Payoff matice pro lov jelena	30
6.3	Payoff matice různých strategií vozítek	31
7.1	Zahajovací rozvaha	34
7.2	Realistický odhad výdajů a zisků	35
7.3	Optimistický odhad výdajů a zisků	35
7.4	Pesimistický odhad výdajů a zisků	36

Tímto bych rád poděkovat především doc. Ing. Mgr. Petru Klánovi, CSc. za vedení a konzultace v průběhu práce. Poděkování také patří mé rodině a blízkým za podporu a pomoc při testování.

Prohlášení

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

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

V Praze dne 11. května 2022

.....

Abstrakt

Tato bakalářská práce se zabývá návrhem, implementací a testováním logické hry pro virtuální realitu. Nejdříve je krátce představena virtuální realita a vytvořena analýza desktopových logických her. Pro prvotní koncept hry je vypracován průzkum trhu a nalezen cílový trh. Dále je na základě předchozí analýzy proveden návrh, implementace a testování prototypu nové logické hry, také je vytvořen finanční plán pro další rozvoj. Výsledkem práce je prototyp logické hry pro virtuální realitu s cílem tvorby autonomních vozítek za pomoci logických obvodů. Výsledky této práce umožňují spolu s finančním plánem pokračovat v udržitelném vývoji hry a vytvořit tak logickou zábavně naučnou hru pro virtuální realitu.

Klíčová slova virtuální realita, Unity, herní cyklus, průzkum trhu, logické hry

Abstract

The goal of this bachelor thesis is to design, implement and test virtual reality puzzle game. Firstly virtual reality is briefly introduced with short analysis of desktop puzzle games. For the concept of the game is conducted market reaserch and the target market found. On the basis of previus analysis is created design, implementation and testing of the game prototype, also financial plan is created for further development. The result is prototype of virtual reality puzzle game. game objective is creation of autonomous vehicles with logic circuits. Results of this thesis enables along with financial plan to continue sustainable development of educational puzzle game for virtual reality.

Keywords virtual reality, Unity, gameplay loop, market research, puzzle games

Seznam zkratk

VR	Virtualní Realita
HMD	Head Mounted Display
VRAC	Virtual Reality Aided Design
KSP	Kerbal Space Program
AR	Augmented Reality
MD	Man Day

Úvod

Virtuální realita nabízí nové zážitky a příležitosti. Hráči přináší nové způsoby interakce, které dokáží být bližší reálnému světu než jsme zvyklí z klasických počítačových her. Díky tomu může být ovládání hry více intuitivnější a zábavnější. Trh logických her pro virtuální realitu je stále v raných stádiích a nabízí mnoho příležitostí, kde se prosadit.

Bakalářská práce je zaměřena na logické hry se stavebnicovým systémem, protože dávají hráči volnější představivost a může tak existovat více řešení jednoho problému. Téma jsem si zvolil, protože existuje pouze jediná logická hra se stavebnicovým systémem pro virtuální realitu Fantastic Contraption. Má ovšem velmi omezený výběr dílů a nenabízí možnost přidání logiky pro vozítka. Rád bych tak vytvořil novou logickou hru, která by řešila problémy hry Fantastic Contraption, dále bych tuto práci rád využil pro další rozvoj hry.

Výsledkem této práce bude prototyp hry ve které bude cílem vytvořit pomocí stavebnicových bloků a logických obvodů jednoduché autonomní vozítka, které bude schopno se samo dostat do cíle. Dále bude výsledkem analýza cílového trhu a návrh finančního plánu pro udržitelný rozvoj nové logické hry.

Práce je rozdělena na několik kapitol. V první kapitole je krátce představena virtuální realita a vytvořena analýza desktopových logických her. V druhé kapitole je proveden průzkum trhu. Ve třetí jsou představeny použité technologie při implementaci, jako jsou Unity, Blender a OpenXR. Ve čtvrté kapitole je proveden návrh, implementace a testování. V předposlední kapitole je krátce představena teorie her a vytvořena payoff matice. V poslední kapitole je staven finanční plán na základě průzkumu trhu pro udržitelný rozvoj hry.



Kapitola 1

Cíl práce

Tato práce má čtyři hlavní cíle. Tyto cíle vycházejí z prvotního konceptu logické hry pro virtuální realitu, ta by měla sloužit jako představení stavby jednoduchých autonomních vozítek za pomoci logických obvodů. První cíl je analyzovat podobné desktopové logické hry a navrhnout metodiku pro jejich rozšíření do virtuální reality.

Druhým cílem je provést průzkum trhu a nalézt cílovou skupinu uživatelů pro novou logickou hru.

Třetím, na základě předchozí metodiky je vytvořit návrh nové logické hry, implementovat návrh logické hry s klíčovými herními mechanikami a otestovat ji.

Posledním cílem je navrhnout finanční plán pro udržitelný rozvoj a šíření hry.

Virtuální realita a logické hry

V této kapitole je krátce představena virtuální realita, její historie a využití. Dále je provedena analýza vybraných logických desktopových her, které souvisí s prvotním konceptem hry. Na základě této analýzy bude později vytvořena metodika rozšíření logických her do virtuální reality.

2.1 Virtuální realita

Pojem virtuální realita můžeme chápat velmi široce. Dal by se vyložit také jako umělá skutečnost. To znamená, že v úplném jádru virtuální reality leží myšlenka uměle vytvořené, vymyšlené reality. Mohly by do ní tedy patřit knihy, nebo dokonce i malby, v zásadě vše co fabrikuje novou realitu. Ovšem dnes se na virtuální realitu spíše díváme z technologického pohledu. Nahrazováním nebo klamáním smyslů se snažíme co nejvíce přiblížit realitě. Z toho důvodu je konečným cílem virtuální reality nahrazení všech smyslů a úplné ponoření do umělé reality.

2.1.1 Historie

Za první nahlédnutí do VR (virtuální reality) lze považovat popsání stereoskopické vize a následné vynalezení stereoskopu Charlesem Wheatstonem v roce 1838. O šest lepozději David Brewster původní design stereoskopu vylepšil nahrazením zrcadel za optické hranoly. V roce 1851 díky vyvinutí tisku obrazů Oliver Wendell úspěšně vytvořimobilní stereoskop. Tím započala nová éra vnímání obrazů, oklamáním stereoskopické vize se rozšířil náš svět o virtuální 3D realitu. [22]

Dalším již technologicky pokročilejším systémem je první multisenzorický simulátor Sensorama. Tento simulátor reality byl poprvé představen roku 1962 Mortonem Heiligem. Systém divákovi umožnil sledovat barevné 3D video doplněné binaurální zvukem, vůněmi, vibracemi a prouděním vzduchu. Jednalo se tedy o první systém s charakteristickými znaky VR systému [7]. Nicméně Sensorama nebyla interaktivní, proto by se mohla spíše považovat za předchůdce dnešních 5D kin, které zážitek většinou rozšiřují o další efekty.

V roce 1965 Ivan Sutherland popsal tzv. The Ultimate Display, který zahrnoval interaktivní grafiku, silovou zpětnou vazbu, zvuk, pach i chuť [7]. O tři roky později představil Sword of Damocles [4]. Displej připevněný k hlavě (HMD), který sledoval pozici hlavy v prostoru a aktualizoval displeje podle této pozice. Rozšířil tak původní myšlenku The Ultimate Display. Sword of Damocles pomocí poloprůhledných zrcadel a dvou mini displejů promítal do vize uživatele počítačové stereoskopické obrazce. [7]

S jinou metodou VR systému přišla laboratoř elektronické vizualizace na Illinoiské vysoké škole v Chicagu. Ta představila na SIGGRAPH konferenci v roce 1992 systém Cave. Cave je

místnost, ve které se na zdi promítá stereoskopický video výstup. Toho je docíleno synchronizováním frekvencí výměny stereoskopických projekcí a obnovovací frekvence projektoru. V kombinaci s využitím závěrkových brýlí docházelo k vytvoření efektu 3D iluze. [15]

Palmer Luckey byl prvním, kdo úspěšně popularizoval VR HMD systém v komerčním sektoru. V roce 2012 spustil crowdfundingovou kampaň na platformě Kickstarter pro svůj HMD Oculus Rift a vybral více než miliony dolarů[21]. O čtyři roky později byla vydána první spotřebitelská verze. HMD měl rozlišení celkové rozlišení 2160x1200 pixelů a obnovovací frekvenci displejů 90 Hz. Pomocí infračervených sledovacích stanic systém dosáhl velmi přesné sledování pozice HMD v prostoru. [20]

2.1.2 Využití VR

Technologie VR se v současné době nejčastěji využívá pro hraní her či jako zdroj dalších zajímavých zážitků. Má ovšem i mnohem širší využití.

VR je často využívána jako nástroj pro vizualizaci při designu, tzv. VRAC (Virtual Reality Aided Design). V automobilovém průmyslu se s VRAC můžeme setkat například u ergonomické analýzy auta, nebo při simulaci montáže. I přesto, že VRAC není tolik přesný a realistický jako klasické způsoby návrhu, představuje výhody v podobě snížení nákladů a zrychlení návrhového procesu. [37]

Další možností aplikace VR jsou trenažéry. Vývoj vojenských leteckých simulátorů byl jedním z důvodů pokroku technologií VR, proto jsou VR trenažéry častou součástí výcviku vojenských pilotů. Díky této technologii se snižuje riziko nehody, výše nákladů a rovněž se zkracuje doba výcviku. [7]

Popularitu si VR v posledních letech získává také ve fyzioterapii, kde bývá kombinována s individuální kinezioterapií. Výhodou je, že odvádí pozornost pacienta, a přispívá tedy ke snížení vnímání bolesti. Tento způsob terapie je rovněž pro pacienty více zábavný, čímž podporuje participaci pacientů v procesu rehabilitace. [30]

Může se také jednat o formu vzdělávání. Příkladem je i hra v této práci, která představuje základy logických hradel a tvorbu jednoduchých interaktivních obvodů.

2.2 Logické hry

Definovat žánr logických her je poměrně složité. Většina her využívá různé úrovně logiky, případně se v jednotlivých segmentech samotné hry objevují logické prvky, např. hádanky. Hra přesto nemusí být považována za logickou. Dle mého názoru je tento žánr velmi komplexní, a proto jej budu definovat podle Mora-Cantalopse. Ten je definuje jako hry, kde řešení problémů je jejich jádrem a hlavní herní mechanikou, řešení těchto problémů je zároveň hlavním zdrojem satisfakce [17]. V následujících částech je provedena analýza dvou logických her, které byly vybrány, protože sdílí prvky mého konceptu. Cílem mé hry je seznámení se základy logických hradel a logických obvodů pomocí stavby autonomních vozítek. Na základě této analýzy dále pak bude zpracována metodika rozšíření logických her do VR a vytvoření návrhu nové VR hry.

V rámci analýzy krátce představím hru, stanovím její cíle a sestavím základní herní cyklus hráče. Obě hry mají systém, který zprostředkovává stavění a ovládání vozidel. Ten je důležité dekomponovat, protože je jednou z hlavních součástí herního cyklu pro tento typ her.

Herní cyklus je jádrem zážitku ze hry. Patří do něj repetitivní akce, které hráč vykonává [8]. Cykly mohou být propojeny a nebo hra může obsahovat i více nezávislých cyklů. Identifikuji hlavní komponenty a přechody cyklu, dle kterých sestavím diagram.

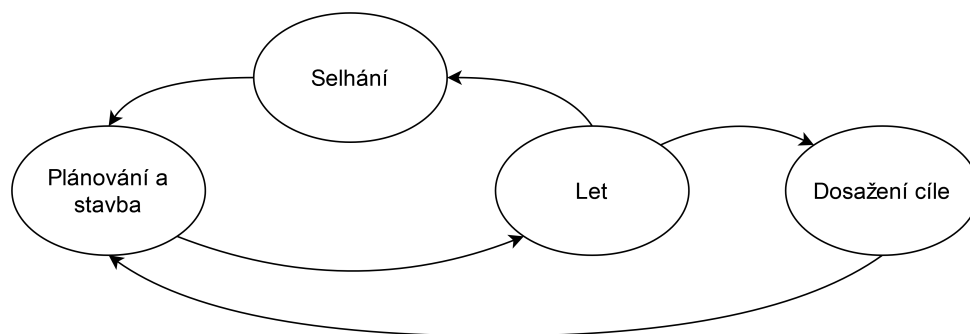
2.2.1 Kerbal Space Program

Kerbal Space Program (KSP) je konstrukční simulátor vesmírných lodí, kde hráč převezme kontrolu nad vesmírným programem planety Kerbin v planetárním systému Kerbol. Může stavět rakety, letadla či jiná vozidla a dobývat vesmír. Hra nemá přímo specifikovaný cíl, záleží na hráči jaké cíle si sám vytvoří [26]. Cíle se také mohou odvíjet výběrem jedním ze tří módů:

- sandbox mód – hráč má odemknutá všechny vylepšení, budovy a díly.
- vědecký mód – na začátku má hráč velmi omezený výběr dílů a možností. Pokrok je zprostředkován vědeckými body, které hráč získává splňováním milníků a vědeckými experimenty.
- kariérní mód – je rozšířením vědeckého módu, přidává ekonomické aspekty, správu rozpočtu a nábor posádky.

Díky otevřenosti KSP není možné stanovit přesný cíl hry. Každý bude hrát hru trochu jinak. Můžeme alespoň stanovit dva obecné cíle. Častým důvodem hraní KSP bývá explorační, tedy zkonstruování rakety a dosažení jiné planety, měsíce či jiného specifického místa. Tento styl hraní bývá spojován se sandbox módem. Dalším stylem hraní je pokoření stanovené hranice nebo dosažení jinak komunitou uměle vytvořeného cíle. S tímto stylem může být spojován vědecký a kariérní mód, kde módy hráči omezují možnosti a dělají tak hru těžší.

Výše zmíněná definice logických her dovoluje zařadit KSP do tohoto žánru, i přesto že není považována za logickou hru. Sestavením herního cyklu KSP je možné ukázat, že patří mezi logické hry. Prvním krokem pro hráče je definování si vlastního cíle (problému). Tímto si zároveň vytváří požadavky na raketu (posádka, náklad, množství paliva, výkon motorů...), tu dle svého návrhu postaví a může se s ní pokusit o dosažení cíle. Jednou z důležitých součástí herního cyklu v KSP je selhání, může být způsobeno mnoha problémy, kterým hráč během letu čelí. Pokud hráč uspěje, dosáhne svého cíle, čímž vyřeší problém, který si sám definoval. S těmito prvky jsem vytvořil následující herní cyklus.



■ **Obrázek 2.1** KSP herní cyklus

Z cyklu lze hru rozdělit na dvě části plánování/stavbu a let. Zaměřím se na část plánování a stavby, protože cílem analýzy je získat důležité informace pro můj koncept. V této části musí hráč navrhnout vhodné řešení problému a realizovat jej. Proto by hra měla poskytovat vhodné nástroje pro plánování a tvorbu. Za pomoci následujícího obrázku představím jednotlivé nástroje, které hra poskytuje.



■ Obrázek 2.2 Ukázka z editoru KSP

1. panel výběru dílů, které jsou rozdělené do skupin dle využití
2. pomocí myši je možné upevňovat vybrané díly přímo na trup
3. zeleno-černé koule znázorňující spojovací kotvy, pokud se k sobě přiblíží, vybraný díl se zacvakne kotvami do sebe
4. výběr symetrie zrcadlení pro upevňování
5. přesnější možnosti manipulace s díly – posun, otáčení a změna rodičovského dílu
6. zdroje informací – těžiště, vztlkový vektor, celkové Δv (změna rychlosti), palivo...
7. správa raketových stupňů a jejich Δv
8. možnost přiřazení akcí (zážeh motoru, vysunutí solárních panelů...) ke klávesám

Základnem plánování je správa raketových stupňů, která umožňuje rozdělit raketu na jednotlivé části určené specifickému problému (dosažení stratosféry, orbity, přistání...). Dále také poskytuje informace o Δv jednotlivých stupňů. Plánování průběhu mise je dále rozšířeno funkcemi z bodů 6 a 8.

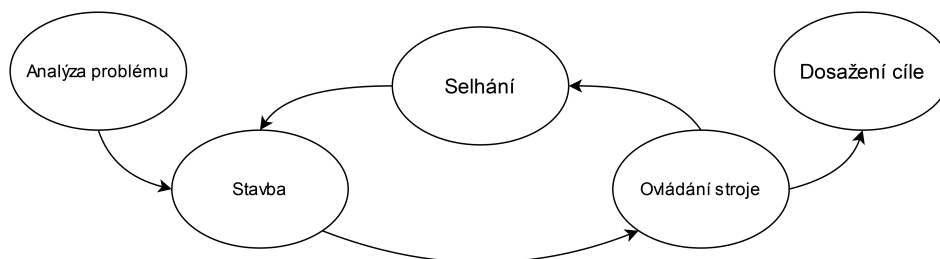
Editor umožňuje sestavování dvěma způsoby. Prvním je připevňování pomocí zeleno-černých kotev, které mají určené pozice a normály na jednotlivých dílech. Pokud se kotva uchopeného dílu přiblíží ke kotvě jiného dílu, tak se uchopený díl se orientuje, že kotvy sdílí stejnou pozici a normály jsou si navzájem opačné. Druhým způsobem je volné upevňování přímo na ostatní součástky. Editor nabízí další rozšiřující nástroje zmíněné v bodech 4 a 5.

Hra poskytuje velmi dobré nástroje pro první část herního cyklu. Uživatelské rozhraní hráči zobrazuje jako první nejdůležitější informace pro tvorbu vozidel (Δv , těžiště, vztlkovou sílu). Dále pak také umožňuje pokročilejším hráčům získat podrobnější informace. Editor pro tvorbu je navržený podobným způsobem. Proto začínající hráči nebudou přehlceni nadměrným množstvím informací. Zacvakávání a připevňování dílů funguje velmi intuitivně a jednoduše. U složitějších vozidel vzniká často problém s přesným umístěním součástí, kde uchopená součástka při mírném pohybu myši přeskakuje mezi různými spojovacími kotvami. To je způsobeno tím, že umístění dílů je zprostředkováno projekcí kurzoru myši do 3D prostoru.

2.2.2 Besiege

Besiege je fyzikálně založená logická hra, kde hráč staví obléhací stroje či jiná vozítka. Hra nabízí 54 úrovní, kde v každé je potřeba splnit úkol nebo vyřešit jednoduchý hlavolam. Cílem hry je tedy vyřešení všech úrovní. Úkoly jsou velmi různorodé od dobývání hradu, přemísťování zásob až krádeže artefaktu. [31]

Herní cyklus bude velice podobný s hrou KSP. Dvěma hlavními rozdíly je, že průběh hry v Besiege je více lineární, po dosažení cíle se hráč posouvá do další úrovně. Druhým je, že analýza problému a plánování nejsou tolik propojené s tvorbou stroje. Prvním krokem pro hráče je analyzování problému, na základě kterého postaví stroj, se kterým se následně může pokusit o splnění cíle. Pokud selže navrátí se do editačního módu, kde může stroj upravit a následně opakovat pokus. Cyklus bude vypadat následovně.



■ Obrázek 2.3 Besiege herní cyklus

Stejně jako u KSP je možné vidět, že hra je rozdělena na dvě části, stavbu a ovládání stroje. Ovládání stroje je zajištěno pomocí přiřazování aktivačních kláves jednotlivým dílům. Lze například přiřadit klávesy motorizovaným kolečkům odděleně každé straně, a tím ovládat vozidlo podobně jako tank. V následujícím obrázku jsou uvedeny důležité nástroje v editoru.



■ Obrázek 2.4 Ukázka z editoru Besiege

1. panel výběru dílů, které jsou rozdělené do skupin dle využití
2. základní blok
3. vybraný díl je kurzorem upevňován ke kotvě bloku
4. nastavení zrcadlení
5. posun a rotace celého stroje
6. nastavení interaktivních bloků – přiřazení kláves, rychlost, zrychlení...
7. přehled skupin bloků přiřazených ke klávesám

Sestavování stroje začíná od základního zeleno-černého bloku. Připevňování funguje podobně jako u kotvového systému v KSP. Rozdílem je, že uchopený díl se nezacvakává dle vzájemné vzdálenosti kotev, ale podle kurzorem vybrané plochy na bloku. Uchopený díl má definovanou jednu hlavní kotvu, kterou je díl připevněn k vybrané kotvě. Před upevněním je zobrazován průhledný náhled upevňované pozice bloku a pokud dochází ke kolizi s ostatními bloky, průhledný náhled zčervená. Zároveň lze bloky při náhledu rotovat kolem normály kotvy.

Systém ovládání nabízí možnost jednoduché logiky a automatizace. To je zajištěno logickými bloky, které mohou při určitém vstupu simulovat stisk klávesy. Nelze oddělit simulovaný stisk klávesy a opravdový stisk klávesy, tudíž je systém limitován množstvím kláves. Ty fungují jako binární propojení mezi bloky. Příklady logických bloků:

- logické hradlo – simuluje hradla AND, OR, XOR...
- senzor – detekuje objekty
- úhломěr – detekuje vychýlení úhlu otočení z nastaveného minima a maxima.

Editor funguje lépe než v KSP, odstraňuje problém nestability způsobené více aktivními kotvami. Hra nabízí přidání logiky a automatizace stroje, ale systém je omezen klávesami a binárními vstupy/výstupy. Logické problémy jsou zajímavé a zábavné. Navíc hra je doplněna zajímavou středověkou atmosférou.

2.2.3 Logické hry ve VR

I pro VR existují logické hry, ale jen jedna sdílí podobné prvky jako koncept mé hry. Fantastic contraption je logická hra, ve které je má hráče postavit vozítko, s nímž následně přemístí červenou kouli do cíle. Hra pracuje na velice jednoduchých principech a dává k dispozici hráči pouze tři základní součástky. Rotující válec, který slouží pro pohon vozidla. Pevnou tyč jako základ konstrukce a volnou tyč která není pevně ukotvena (může se v připevněném bodě otáčet) [18]. Přestože hra poskytuje pouze tři základní díly, je možno postavit zajímavá vozítka. Vozítko nelze nijak ovládat, po odstartování úrovně se jen aktivují rotující válce.

Pozice hráče ve hře je přímo spojená s reálnou pozicí, není tedy umožněno se posouvat ve VR. To představuje jistá omezení. Hráč se musí více ohýbat a zaklánět hlavou, delší čas strávený ve hře může zatěžovat krk a způsobovat tak bolesti. Dalším problémem editoru je, že umísťování součástek je volné. Je náročné součástky zarovnat do roviny, nebo vytvářet symetrické tvary. Editor je velmi jednoduchý a intuitivní, ale chybí v něm pokročilejší nástroje.

Průzkum trhu

Cílem průzkumu trhu je identifikovat cílový trh a určení důležitých informací spojených s produktem (akceptovatelná cena, konkurence, objem prodeje). Průzkum trhu se dělí na dvě části na primární a sekundární. Primární průzkum je aktivní přístup k potenciálním zákazníkům (dotazníky, pohovory, pozorování). Sekundární průzkum je získávání dat již z existujících zdrojů. Z časových a finančních důvodů, se tak často začíná sekundárním průzkumem [16]. Tato práce je zaměřena na sekundární průzkum.

Nejdříve musíme určit celkový trh. Ten Srpová definuje následovně: „Celkový trh zahrnuje všechny myslitelné možnosti využití daného výrobku nebo služby.“ [32, s. 19] Celý trh se podrobně nepopisuje, vymezí se zejména na skupiny zákazníku, které dle Srpové [32]:

- mají z výrobku nebo služby značný užitek
- mají k výrobku či poskytované službě snadný přístup
- jsou ochotny za produkt či službu zaplatit

Z celkového trhu je nutné se zaměřit na konkrétní potřeby zákazníků, proto se takto vymezený celkový trh dále rozdělí na menší segmenty. Dělení segmentů se odvíjí od daného trhu a produktu. Dělit se dá například dle potřeb zákazníků. Z takto segmentovaného trhu je potřeba vybrat segmenty nejvhodnější pro náš produkt. Kriteria pro výběr segmentu bude především výše možného zisku, ale např. také velikost a růst segmentu, výhody oproti konkurenci, shoda produktu s potřebami zákazníků. Takto vybrané segmenty tvoří hledaný cílový trh. [32]

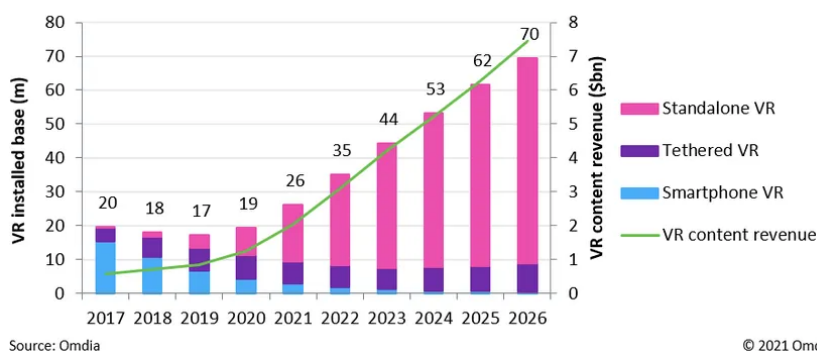
Dále k cílovému trhu a danému produktu je potřeba stanovit důležité otázky, na které v rámci průzkumu trhu nalezneme odpověď [32]. Častými typy otázek jsou např. akceptovatelná cena, zájem o produkt, kde zákazníci nakupují.

3.1 Celkový trh

Celkový trh pro VR hru vymezíme pomocí skupiny, která má k produktu jednoduchý přístup. To budou potenciální zákazníci, kteří vlastní moderní VR HMD. Tato skupina tvořila v roce 2020 29,7 % z globálního VR trhu [33]. Dle zprávy Omidia v roce 2021 prodej VR HMD dosáhl 12,5 milionu jednotek a příjmy z obsahu pro VR činili 2,1 miliard dolarů [12]. Následující graf zahrnuje historická data a projekce až do roku 2026.

Trh s VR HMD má velký růstový potenciál. Ve zprávě Omidia je penetrace trhu v roce 2021 stanovena na 2,4 HMD na sto domácností. V roce 2026 je předpokládaná penetrace 6,3 HMD na sto domácností [12].

Consumer VR headset active installed base and content revenue, 2017-2026



Source: Omdia

© 2021 Omdia

■ **Obrázek 3.1** Projekce a historická data prodeje VR HMD a příjmů z obsahu pro VR [5]

Z grafu je největším a nejrychleji rostoucím segmentem standalone VR (HMD, které nepotřebuje připojení k pc). Tento segment ovšem přináší svá omezení, výpočetní výkon standalone VR je omezenější, tudíž je třeba hry pro tento typ zařízení optimalizovat. Druhým největším segmentem je tethered VR (HMD, které vyžaduje připojení k pc). Díky připojení k pc má tento typ HMD zpravidla dostupnější vyšší výkon. Nevýhodou je vyšší pořizovací cena a potřeba výkonného pc. Poslední již ustupujícím segmentem je mobilní VR, které jako hlavní součást HMD využívá mobilní telefon. Je to speciální případ VR HMD, které často nemá ovladače a je spíše používáno jako moderní stereoskop.

3.2 Cílový trh

V předchozí části jsme určili hlavní segmenty celkového trhu, dle typu HMD. Vývoj pro tethered VR, je jednodušší díky vyššímu dostupnému výkonu. Ovšem vynechání standalone VR, by značně omezilo dostupný trh a především příležitost na trhu. Možnou strategií je prvně vydat produkt pro menší segment trhu tethered VR a v rámci dalšího vývoje expandovat na trh se standalone VR.

Pro určení cílového trhu je třeba definovat přesněji produkt. Produktem je VR logická hra s prvky fyzikální simulace. Hra je inspirována desktopovými tituly (Kerbal Space Program a Besiege) a vozítka autonomus line follower (typ autonomního vozítka, které sleduje čáru). Cílem hry je postavit z dostupných bloků a logických hradel autonomní vozítko, které se dostane co nejrychleji do cíle. Jedná se tedy o logicky náročnější hru a která slouží jako zjednodušené představení tvorby logických obvodů.

Cílem produktu tedy bude VR herní trh, ten můžeme dále segmentovat dle jednotlivých žánrů (logické hry, střílečky, simulátory). Je zřejmé, že produkt bude spadat do segmentu logických her, ten dále vymezíme nejpodobnějším produktem, hrou Fantastic Contraption. Cílovým trhem jsou hráči Fantastic Contraption a sekundárním cílovým trhem jsou hráči logických VR her.

3.3 Sekundární průzkum

Pro průzkum trhu je třeba stanovit důležité otázky a nalézt pro ně odpovědi. Cílem sekundárního průzkumu je nalezení co nejvíce odpovědí a odhadů na ty to otázky pomocí již dostupných informací. Stanovené otázky:

- velikost cílového trhu

- růst segmentu
- akceptovatelná cena
- co chybí zákazníkům u konkurence
- očekávání zákazníků od produktu

Velikosti cílového trhu určíme jako počet prodaných kopií Fantastic Contraption. Oficiální data prodaných kopií nejsou zveřejněny, je potřeba provést odhad. Dle článku v Upload VR v roce 2017 Fantastic Contraption vygenerovalo více jak milion dolarů [10]. Při současné ceně dvaceti dolarů za kopii stanovíme dolní hranici prodaných 50 000 kusů. Na základě grafu ze zprávy od Omidia můžeme do odhadu zahrnout růst trhu. Od roku 2017 se příjmy z obsahu pro VR zčtyřnásobili [12]. Cílový trh nemusel růst stejně rychle jako samotný celkový trh, proto vytvořím dva odhady, realistický a optimistický. Pro realistický budeme předpokládat 50 % růst prodeje a pro optimistický 100 % růst prodeje od roku 2017. Celkem máme tři odhady pro velikost cílového trhu:

- dolní hranici odhadu (pesimistický odhad) – 50 000 prodaných kusů
- realistický odhad – 75 000 prodaných kusů
- optimistický odhad – 100 000 prodaných kusů

Růst cílového segmentu bude spíše omezený, z důvodů minimálního rozvoje a marketingu Fantastic Contraption, dá se ovšem předpokládat, že růst sekundárního trhu (žánr logických her) bude stejný jako celkového trhu.

Kupní síla celkového trhu je silná, pořizovací cena základního VR HMD je poměrně vysoká (přibližně 10 000 Kč). Akceptovatelnou cenu produktu stanovíme porovnáním s podobnými produkty. Ceny současně populárních logických her jsou následovné dle obchodu Steam [19]:

■ **Tabulka 3.1** Ceny populárních her

Hra	Cena
Fantastic Contraption	490 Kč
The Talos Principle VR	900 Kč
Hyperbolica	340 Kč
The Room VR: A Dark Matter	730 Kč
Myst	610 Kč

Ceny jsou mezi 340–900 Kč, ty se odvíjí od propracovanosti samotné hry. Hledaná akceptovatelná cena pro produkt bude srovnatelná s hrou Fantastic Contraption. Tu stanovíme na 450–650 Kč.

Hlavní konkurencí je Fantastic Contraption, z recenzí uživatelů na platformě Steam, lze analyzovat co uživatelům chybí. Jako hlavní problémy dle recenzí jsou [18]: nedostatek dílů, nelze ovládat vozítka, nedostatek variací, neintuitivní UI.

Pro stanovení očekávání zákazníků od produktu, je třeba provést primární průzkum spojený s daným produktem. Částečně lze očekávání určit jako napravení nedostatků hlavní konkurence.

Použité technologie

Cílem této kapitoly je krátké seznámení s použitými technologiemi a terminologií při vývoji prototypu hry.

4.1 Unity

Engine Unity jsem zvolil, protože je flexibilní a mám s ním již předchozí zkušenosti. Je to multiplatformní herní engine s komponentovou architekturou herních objektů. Je průběžně rozšiřován společností Unity Technologies, která vydala první verzi v roce 2005. Unity se vyznačuje svojí univerzálností, umožňuje vývoj 2D a 3D aplikací pro většinu dostupných platforem (Windows, Linux, Android, Google Stadia...). Vývoj v engineu je zprostředkován grafickým rozhraním Editoru a tvorbou skriptů v programovacím jazyce C#. Při vývoji je hra spouštěna přímo v editoru a je možno za běhu využívat většinu nástrojů. Výhodou Unity je licence zdarma pokud celkový zisk nebo financování je menší než 100 000 \$ za posledních 12 měsíců. [36]

4.1.1 Inspektor

Inspektor je důležitou částí grafického rozhraní. Používá se pro editaci herních objektů, přidávání komponentů a jejich konfiguraci [34]. Výhodou inspektoru je možnost jeho úpravy, například pomocí PropertyDrawer třídy se dá specifikovat vlastní zobrazení třídy [27], toho je využito v této práci pro zobrazování a editaci bitů třídy snibble.

4.1.2 Prefab a Prefab Variant

Prefab systém unity funguje jako šablona pro instancování nových herních objektů. Prefab Asset je uložený herní objekt na který se odkazují skripty pro tvoření instancí Prefabu. Systém podporuje vnořené Prefaby a Prefab Variants. Prefab Variants jsou upravenou verzí Prefab Assetu, fungují podobně jako dědění tříd. Změny v hlavním Prefab Assetu propagují do Prefab Variants. To se hodí pro tvorbu mnoha variant podobných objektů, které se liší např. jen částí konfigurace komponentů. [6]

4.1.3 Komponenty

Komponenty jsou funkční části herních objektů. Každý herní objekt může mít více komponentů, ale každý komponent má jeden Transform komponent (určuje pozici, rotaci, měřítko objektu

a propaguje změny na podobjekty). Jsou přiřazovány v inspektoru a často se využívají k změně chování a vlastností herního objektu. [11]

4.1.3.1 Rigidbody

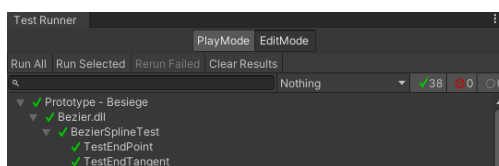
Rigidbody komponent přidává herní objekt do fyzikální simulace. Takto je možné simulovat reálné kolize s jinými objekty a manipulovat s nimi pomocí aplikování síly nebo točivého momentu. Komponent má nastavení pro změnu chování např. vypnutí gravitace. Vlastnost Is Kinematic dovolí změnu pohybu jen pomocí komponentu Transform, tudíž fyzikální síly nezmění pozici objektu. [28]

4.1.3.2 Fixed Joint a Configurable Joint

Jsou komponenty, které zajišťují fyzikální spojení mezi dvěma objekty s Rigidbody komponentem. Fixed Joint je pevné spojení, které dovoluje minimum pohybu. Spojením více propojení za sebou způsobí ohyb ve spojeních, který je způsoben stylem fyzikální simulace. Configurable Joint odhaluje různá nastavení, pro vytvoření komplikovanějších způsobů projení např.: pantový ohyb, rotace, pružinový spoj. [13]

4.1.4 Test Framework

Test Framework je nástroj pro automatizované testování samostatného kódu a kódu který závisí na herních instancovaných objektech nebo komponentech, které musí být umístěny v běžící scéně. Testy jsou definovány v samostatné C# Assembly definici, která definuje jaké balíčky testy budou využívat. [35]



■ Obrázek 4.1 Ukázka testovacího nástroje

4.2 OpenXR

OpenXR je standard pro komunikaci mezi VR/AR zařazeními a softwarem. Cílem je sjednocení komunikace a odstranění problému fragmentace softwaru pro dané zařízení. To je docíleno API vrstvou OpenXR. OpenXR se do Unity instaluje formou pluginu. Plugin přidá komunikaci mezi zařízeními a Unity Action Input systémem. [23]

4.3 Blender

Blender je open-source 3D grafický program s širokým výběrem nástrojů. Primárně jsem Blender využíval pro tvorbu modelů. Mezi funkce blenderu patří texturování, vfx, animování, tvorba modelů a další. [3]

Návrh, implementace a testování

V této kapitole je proveden návrh herních systémů na základě předchozí analýzy logických her. Návrh se bude soustředit na editor a logický systém. Dále je přiblížena implementace navržených herních systému a způsoby testování při implementaci.

5.1 Návrh

Cílem návrhu je vytvořit koncept herních systémů, které budou tvořit jádro hry. Logická hra by měla sloužit jako představení základu práce s logickými hradly a návrhu jednoduchých logických obvodů. Herní cyklus by měl být podobný s herními cykli úspěšných her KSP a Beseige, jsou příkladem logických her, kde cílem je postavit vozítko/stroj řešící problém. V analýze těch to her byli identifikovány dvě důležité části tvořící herní cyklus: stavba a simulace (průběh řešení). Je potřeba navrhnout vhodné nástroje (editor) pro stavbu vozítka a logický systém pro ovládání.

5.1.1 Metodika rozšíření do VR

Rozšiřování desktopových her do VR se bude velmi odvíjet od samotné hry. Některé hry budou velmi jednoduché rozšířit a jiné budou s klasickými metodami obtížné např. ovládání strategických her pro VR bývá odlišné od desktopových her. Při rozšíření her do VR je nutné myslet na rozdílný styl pohybu a interakce.

Nejdříve je vhodné určit pro hru typ lokomoce (pohybu). Důležitým faktorem při výběru pohybu je nevolnost z pohybu. Někteří uživatelé mají problém s nevolnostmi, pokud není pohyb ve virtuální realitě propojen s pohybem v reálném světě. Časté typy lokomoce jsou:

- teleportace – hráč se může teleportovat na vybraná místa, tento typ řeší problém s nevolnostmi
- posouvání – hráč se posouvá pomocí ovladačů, pohyb je plynulý, často vyvolává nevolnost
- world pulling – hráč se posouvá pomocí ovladačů a následného přitažení k sobě, hráč se pohybuje po menších kouskách, trochu pomáhá proti nevolnostem
- fixní pozice – hráč se pohybuje jen v rámci systému sledování pozice (je svázán s jedním místem)

Dále je důležité zvolit jak bude hráč interagovat, zde se lze inspirovat z reálného života. Místo otevření dveří zmáčknutím tlačítka, hráč otevře dveře uchopením a otočením kliky. Kde je to možné, je dobré interakce zakládat na uchopování věcí. Uživatelé ve VR mají tendenci snažit se věci uchopit a manipulovat s nimi jako v reálném životě.

5.1.2 Editor

Je důležité, aby editor pro hráče byl intuitivní a měl vše důležité na dosah ruky. Zároveň by měl dovolit plánovat řešení problému. Proto je vhodné, aby stavba vozidla se odehrávala přímo v dané úrovni (podobně jako v *Fantastic Contraption* a *Besiege*). Stavebnicový systém kotev z hry KSP je vhodným kandidátem, pro stavební systém. Ve hře KSP je při složitějších vozidel tento systém nestabilní, důvodem je promítání kurzoru do 3D prostoru. VR tento problém řeší sledováním polohy ve 3D prostoru a poskytnutí stereoskopické vize uživateli. Součástky tak můžou být spojovány pomocí kotvového systému. Při uchopení dílu se aktivují jeho kotvy a hráč tak tento díl může připevnit k jinému dílu. Uchopení zároveň může sloužit k oddělování dílů od sebe, při uchopení se rozpojí všechny kotvy.

Lokomoce hráče by měla dovolovat přesnější práci se součástkami a volný pohyb v celém 3D prostoru. Zároveň by měla být dostatečně rychlá, aby hráč stihl sledovat vozítko. Jednoduchým způsobem jak docílit těchto vlastností je použití typu lokomoce *world pulling* (chycení a přitažení). Hráč se chytí jednou rukou v prostoru a tou se může posouvat (odstrkovat). Při střídání rukou se hráč může pohybovat relativně rychle, ale zároveň umožňuje jemný pohyb. Tento pohyb se dá dále rozšířit o zmenšování/zvětšování celé úrovně, takto dovolí hráči přímo kontrolovat rychlost a přesnost pohybu. Změna měřítko úrovně může být docílena chycením oběma rukami a jejich přibližováním/oddalováním (podobně jako přibližování obrázku na mobilním telefonu).

Získávání dílu by bylo možné docílit dvěma způsoby. První je otevíratelné menu připevněné k ruce. Druhým způsobem je mít statické objekty, které vytváří nový díl pokud díl odebrán. První způsob je vhodnější pro větší množství dílů a získávání dílů je hráči jednoduše přístupné, ale jeho implementace je složitější. Druhý způsob omezuje množství dílů dostupné hráči a pro získání dílů se musí hráč posunout. Díky možnosti změny měřítko potřeba pro posunutí je snížena. Druhý způsob je pro začátek vývoje vhodnější z důvodu jednoduché implementace, a tak rychlejšímu testování.

Díly by měli být přehledné, velikostně kompatibilní a univerzální. Nejlepším příkladem takovýchto dílů jsou kostičky LEGO. Jsou velmi jednoduché a univerzální, tak umožňují velmi kreativní výtvoř. Hra *Beseige* má podobný stavebnicový systém tvořen převážně z kvádrových a krychlových dílů. Je potřeba mít stanovenou jednotku rozměru a od té odvíjet krychlovité díly. Každý díl (blok) by měl mít jednu hlavní funkci a spadat do jedné ze čtyř kategorií:

- konstrukční blok – zajišťuje více kotevních bodů a tvoří tak kostru vozítka
- logický blok (obvod/hradlo) – na základě vstupu generuje výstup
- výstupní blok – na základě vnějších faktorů generuje výstupů
- vstupní blok – na základě vstupu vykonává akci

5.1.3 Logický systém

Cílem logického systému je zprostředkovávat komunikaci mezi jednotlivými komponenty vozítka. Je potřeba stanovit jakým způsobem tato komunikace bude zprostředkována. Komunikace bude probíhat mezi těmito typy bloků: logický, výstupní, vstupní.

Nejzákladnější komunikační jednotkou je bit, ta je použita ve hře *Beseige* pro základní automatizaci a logiku. Používání bitu ke komunikaci je limitující, neumožňuje jemnější ovládání. Blok poháněného kolečka by potřeboval alespoň dva bity pro ovládání. První bit by určoval zda se kolečko má točit nebo ne a druhý by určoval směr otáčení. Výhodou tohoto přístupu je, že je jednodušší, ale vyžaduje více vstupů.

Další možností je použití bytu ke komunikaci. Pokud by hráč chtěl vytvořit svůj vlastní signál pomocí bitů, každý by musel manuálně napojit, to je velmi pracné pro hráče. Je možné využít půl byte (tvořen 4 bity) též označován jako nibble. To přidává možnost více stupňové přesnější kontroly. Ovšem pro příklad s kolečkem by stále bylo potřeba dvou vstupů pro kontrolu.

Pro eliminaci jednoho vstupu a zjednodušení zapojení můžeme použít nibble se znaménkem tzv. snibble. Kde poslední bit je použit jako označení znaménka (snibble má dvě nuly zápornou a kladnou). Díky použití snibble můžeme pro příklad s kolečkem eliminovat vstup pro směr otáčení. První tři bity budou využity pro ovládání rychlosti (8 úrovní rychlosti) a poslední bit znaménka pro směr otáčení.

Tento logický systém může simulovat komunikaci pomocí bitů. Bitový vstup může akceptovat snibble vstup tak, že nenulový signál považuje za bitovou 1 a nulový signál za bitovou 0. Bitový výstup bude vysílat maximální kladný výstup snibble (0111 – kladné číslo 7) jako bitový signál 1, pro bitový signál 0 bude vysílat nulový snibble. Takto můžou být kombinovány bitové a snibbleové logické bloky. Proto je nevhodnější vybrat logický systém využívající snibble.

5.2 Implementace

V této podkapitole je představena implementace jednotlivých částí hry. Pro implementaci hry bylo využito herního enginu Unity 2020, pluginu OpenXR a modelovacího softwaru Blender. Architektura se odvíjí od samotného enginu. Herní engine pracuje na komponentové architektuře, tudíž většina práce s enginem je zaměřena na tvorbu herních objektů s vlastními a již implementovanými komponenty.

5.2.1 Díly a fyzikální systém

Díly dle návrhu mají mít kotvy, které se při uchopení aktivují a zacvaknou do nejbližší kotvy (pokud je dostatečně blízko). Dále by jednotlivé díly měly simulovat fyzikální kolize a interakce. Herní objekt se přidá do fyzikální simulace unity pomocí komponentu RigidBody. Dalším požadavkem na díly je, aby byli jednoduše upravitelné pro vytváření různých variant dílů.

V různých částech projektu je potřeba upozornit na opuštění editoru a spuštění úrovně. Toho je docíleno použitím observer patternu. Komponent BuildModeBlockObserver zajišťuje u dílů, aby při startu úrovně odemkla fyzika dílů a mohli se začít volně pohybovat. Při navrácení do editoru se fyzika dílů vypne a vrátí se na původní pozici v editoru.

Kotvy zprostředkovává vlastní komponent Endpoints. Ten spravuje herní objekty endpoint (kotva) pro danou instanci dílu. Zajišťuje získávání nejbližšího objektu v rámci spravovaných endpointů a fyzikální propojování/odpojování s ostatními bloky. Spojení dvou dílů dohromady je zprostředkováno komponentem Fixed Joint poskytovaný enginem. Komponent Endpoints také zároveň v inspektoru umožňuje vyplnit pozice a normálový vektor jednotlivých kotev, tudíž tvorba nových bloků s kotvami je jednoduchá rychlá.

FixedJoint spojí vzájemně relativní pozice dvou herních objektů, proto je potřeba nastavit správnou pozici a rotaci herního objektu dílu před propojením. Tuto funkcionalitu zajišťuje vlastní komponent AttachController. Navíc zobrazuje průhlednou kopii dílu pokud existuje poblíž kotva pro umístění. Kopie je umístěna tak jak bude uchopený díl připevněn k druhému dílu, pokud se kopie překrývá s jiným blokem tak zčervená jako indikace neplatného umístění.

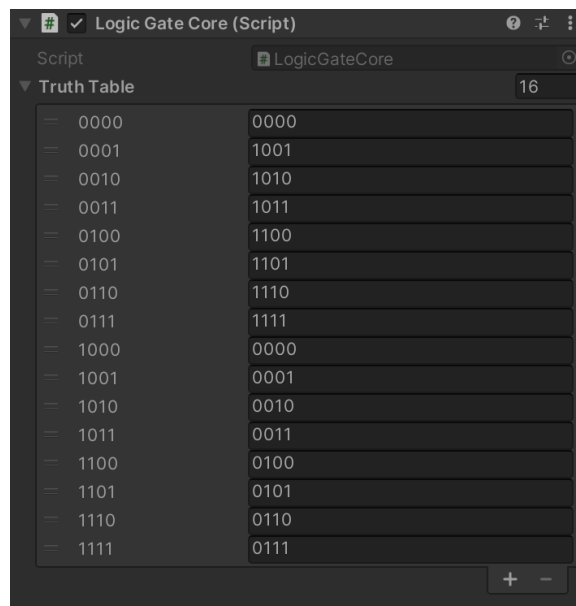
5.2.2 Logický systém

V návrhu je určeno jak by se logický systém měl chovat. Je ale potřeba stanovit další požadavky. Systém by měl podobně jako díly jednoduše rozšiřitelný a modifikovatelný. Hra bude mít celkem

tři typy bloků, které byly stanoveny v návrhu. Podle těchto typů bloků lze logický systém rozdělit na tři části: vstup, výstup a jádro logického obvodu. Vstup a výstup by měli fungovat odděleně od logického jádra, protože mohou být využity jen jako samostatné části pro vstupní/výstupní bloky. Komunikace mezi jednotlivými částmi je zajištěna pomocí observer patternu za cílem dosažení menší provázanosti.

Komponent Receivers zpracovává vstupy z jednotlivých konektorů. Jakmile dostane kompletní vstup (ze všech připojení) pošle observerům zpracovaný snibble vstup. Komponent Transmitters přijme snibble zpracuje a notifikuje signálem observery. Oba komponenty umožňují konfigurovat počet konektorů (maximálně čtyři) a jestli mají přijímat/vysílat jednotlivé bity signálu nebo celý snibble. Jestliže komponent má přijímat snibble místo jednotlivých konektorů pro bity může mít pouze jeden konektor pro snibble. Komponenty zpracovávají vstupy/výstup dle návrhu. Pokud je na vstup posláno více signálů samotný konektor upřednostňuje nejsilnější signál (absolutní hodnotu signálu) a pokud se signály rovnají upřednostní pozitivní signál.

Pro jednoduchou a rychlou tvorbu logických obvodů je využíván komponent LogicGateCore. Ten se přihlásí jako observer ke komponentu Receivers. Přijímaný signál vyhledá v pravdivostní tabulce a podle ní vyšle přes Transmitters signál. Tabulka je omezena na 16 záznamů, a proto Receivers a Transmitters mohou pracovat jen s čtyřmi bity nebo celým snibblem. Pro pravdivostní tabulku bylo potřeba vytvořit vlastní PropertyDrawer, aby v inspektoru zobrazovala signál jako bity.



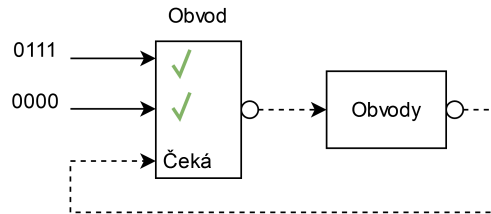
Input (4 bits)	Output (4 bits)
0000	0000
0001	1001
0010	1010
0011	1011
0100	1100
0101	1101
0110	1110
0111	1111
1000	0000
1001	0001
1010	0010
1011	0011
1100	0100
1101	0101
1110	0110
1111	0111

■ **Obrázek 5.1** Ukázka pravdivostní tabulky obvodu pro násobení signálu mínus jedničkou

Za využití tabulky v inspektoru a funkce prefab variant je možné velmi rychle vytvářet nové logické obvody, je potřeba jen nastavit počet vstupů/výstupů, vyplnit pravdivostní tabulku. Další výhodou využívání pravdivostní tabulky jako jádra logického obvodu je možnost implementace jednoduchého rozšíření o runtime editor logických obvodů a jejich následného ukládání. Pro ukládání nově vytvořených obvodů by bylo potřeba uložit jen název obvodu, pravdivostní tabulku a nastavení vstupu/výstupu.

Obvody a jejich zapojení musí tvořit acyklický orientovaný graf, pokud by v grafu vznikl cyklus (kružnice), tak by se obvod dostal do neřešitelného stavu a nevysílal by žádný signál. Důvodem neřešitelného stavu při cyklu, je způsob zpracování vstupu. Komponent Receivers čeká dokud nedostane signál ze všech připojení, až pak zpracuje a pošle signál dál. Cyklus

v obvodech tedy způsobí, že nějaký obvod bude čekat na příjem signálu a příjem toho signálu bude závislý na signálu z toho samého obvodu. Při propojování obvodu je kontrolováno zda propojení nevytvoří cyklus, pokud by mělo vytvořit, hráči nedovolí obvody propojit. Kontrola probíhá pomocí algoritmu hledání do šířky, který prochází graf (obvody), a pokud dojde do počátečního vrcholu (obvodu), tak přidané propojení vytváří cyklus a je nevalidní.



■ **Obrázek 5.2** Příklad neřešitelného stavu způsobeným cyklem

5.2.3 Beziérový křivky

Pro reprezentování propojení obvodů pomocí kabelu je vhodné využít Bézierovy křivky, podél kterých je možno vytvářet mesh kabelu. Unity neposkytuje implementaci těchto křivek, proto bylo potřeba vytvořit vlastní. Obecná definice křivky je dle Pierr Béziera [2] definována následovně:

$$\mathbf{B}(t) = \sum_{i=0}^n \binom{n}{i} (1-t)^{n-i} t^i \mathbf{P}_i$$

Kde \mathbf{P}_0 až \mathbf{P}_n je množina kontrolních bodů a $\mathbf{B}(t)$ je bod na křivce pro parametr $t \in (0, 1)$. Bod \mathbf{P}_0 označujeme jako počáteční a \mathbf{P}_n jako koncový.

Je potřeba uživateli umožnit vybrat kudy kabel povede. Při propojování budou zaznamenávány body s danou vzdáleností. Mezi těmito body je potřeba interpolovat křivku. Nejvhodnější je využít spojení více kubických Bézierových křivek. Pro vytvoření kubické Bézierovy křivky jsou potřeba čtyři body, explicitní forma vzorce je následovná:

$$\mathbf{B}(t) = (1-t)^3 \mathbf{P}_0 + 3(1-t)^2 t \mathbf{P}_1 + 3(1-t) t^2 \mathbf{P}_2 + t^3 \mathbf{P}_3$$

Bod \mathbf{P}_1 je dále označován jako počáteční kontrolní bod a \mathbf{P}_2 koncový kontrolní bod. Označení \mathbf{P}_0 a \mathbf{P}_3 zůstává stejné (počáteční a koncový bod).

Bézier spline je spojení více Bézierových křivek za sebou. Přechod křivek je hladký pokud v bodě přechodu jsou pro obě křivky stejné tečny [9]. Pro kubické Bézierovy křivky je hladkost navázání dvou křivek zjištěna tak, že pro koncový bod \mathbf{P}_3 první křivky a počáteční bod $\bar{\mathbf{P}}_0$ druhé křivky platí $\mathbf{P}_3 = \bar{\mathbf{P}}_0$. Dále koncový kontrolní bod \mathbf{P}_2 a počáteční kontrolní bod $\bar{\mathbf{P}}_1$ dvou křivek leží na přímce tvořené body \mathbf{P}_2 , \mathbf{P}_3 a $\bar{\mathbf{P}}_1$, a zároveň platí že:

$$|\mathbf{P}_2 \mathbf{P}_3| = |\mathbf{P}_3 \bar{\mathbf{P}}_1| \wedge \mathbf{P}_2 \neq \bar{\mathbf{P}}_1$$

Pro interpolaci je k dispozici řada bodů \mathbf{P}_n . Počáteční kontrolní bod a koncový kontrolní bod jsou tvořeny směrovými vektory konektorů. Interpolace mezi sousedními body vytvoří \mathbf{B}_{n-1} kubických křivek, pro ty je potřeba dopočítat chybějící kontrolní body. Kontrolní body pro bod \mathbf{P}_i , $0 < i < n$ se dopočítají následovně:

1. na kolmici k ose úhlu tvořeným body $\angle \mathbf{P}_{i-1}\mathbf{P}_i\mathbf{P}_{i+1}$, která prochází \mathbf{P}_i a ležící v rovině daného úhlu leží kontrolní body.
2. kontrolní body jsou vzdáleny $|\mathbf{P}_{i-1}\mathbf{P}_i|/2$ od bodu \mathbf{P}_i
3. kontrolní bod bližší k bodu \mathbf{P}_{i-1} je koncovým kontrolním bodem křivky \mathbf{B}_{i-1}
4. vzdálenější bod je počátečním kontrolním bodem křivky \mathbf{B}_i

Dále pro generaci meshe je potřeba znát tečnu v bodě křivky, tu získáme derivací křivky v bodě. Pro kubické Bézierovy křivky je derivace následující:

$$\mathbf{B}'(t) = 3(1-t)^2(\mathbf{P}_1 - \mathbf{P}_0) + 6(1-t)t(\mathbf{P}_2 - \mathbf{P}_1) + 3t^2(\mathbf{P}_3 - \mathbf{P}_2)$$

5.2.4 Generování meshe propojovacího kabelu

Při generování meshe je vhodné rozdělovat body na křivkách rovnoměrně dle délky jednotlivých křivek. Délka křivek se musí aproximovat. To je docíleno rozmístěním bodů na křivce a spočítáním zváženosti mezi sousedními body. Toto při generování bodů na Bézierově splině umožní převést parametr $t \in (0, 1)$ na vzdálenost od počátku křivky. Je nutno efektivně zjistit na které křivce bod leží, protože bodů na splině se bude vytvářet mnoho a každému je potřeba přiřadit křivka, která bod vygeneruje. Jednoduchým způsobem jak docílit $\mathcal{O}(\log n)$ časové náročnosti je pomocí modifikovaného binárního vyhledávání, které najde index intervalu vzdálenosti (index křivky) od počátku křivky. Pseudokód kódu intervalového binárního vyhledávání:

■ **Výpis kódu 5.1** Pseudokód modifikovaného binárního vyhledávání

```
// Intervals <0; 2>, (2; 3>, (3, 6>
// will be interpreted as array [0, 2, 3, 6]
//
// Used for finding on which bezier curve is point located by distance

private int BinarySearchIntervalIndex(float num, List<float> intervals){
    // Initial setup of searched indexes
    int low = 0;
    int high = intervals.Count;

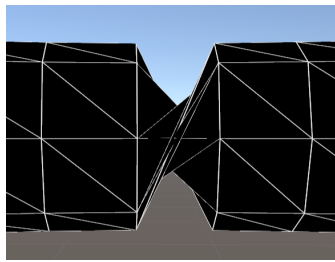
    // Binary search loop
    while(low < high){
        // Middle index of range
        int mid = Mathf.FloorToInt((low + high) / 2);

        // Move range
        if(intervals[mid] < num && mid != low){
            low = mid;
        }

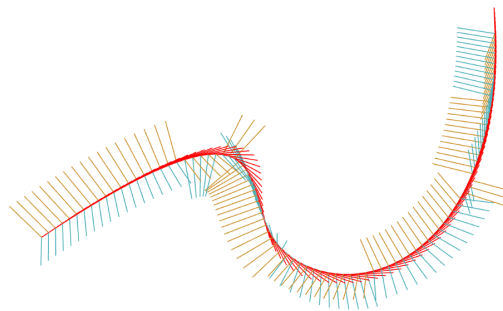
        //Move range
        else if(intervals[mid] >= num && mid != high){
            high = mid;
        }
        //Found index, end loop
        else{
            low++;
            high = low;
        }
    }
}
```

```
// For first interval return 0
if(low == 0){
    return 0;
}
// Return interval
return low - 1;
}
```

Při generování bodů vzniká další problém. Normálové vektory nejsou jednoznačně určeny, tudíž můžou nepravidelně rotovat kolem křivky a způsobovat překrucování meshe.

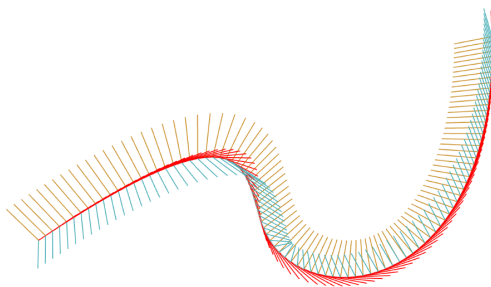


■ **Obrázek 5.3** Ukázka chyby generování meshe způsobené rotací normálových vektorů



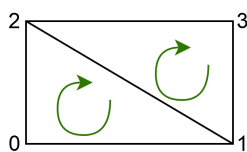
■ **Obrázek 5.4** Příklad náhlých rotací normálových vektorů [1]

Pro minimalizování rotace normál je využita projekční metoda navrhnuta Fopke Klokem v roce 1986 [14]. Ta diskrétně aproximuje minimální rotace normálových vektorů tak, že počáteční normálový vektor projektuje na normálovou rovinu následujících bodu. Směr projekce je určen prvním a následujícím bodem. Takto se pro každý generovaný bod od začátku do konce vytvoří normálové vektory s minimalizovanou rotací. Důvodem vybrání tohoto diskrétního přístupu je jednoduchá implementace např. oproti metodě dvojité reflexe. Klokova metoda není přesná, ale mírná rotace v tomto případě nevádí.

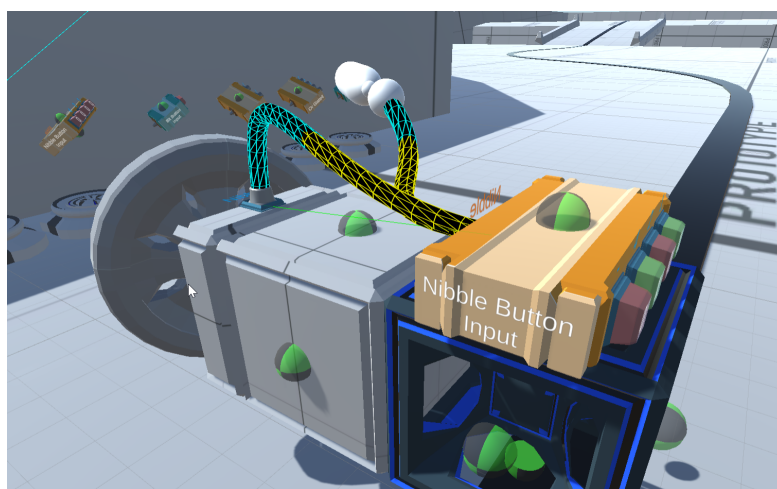


■ **Obrázek 5.5** Příklad minimalizovaných rotací normálových vektorů [29]

Samotné generování meshe probíhá tak, že pro daný počet bodů na křivce se vygenerují vrcholy meshe podle uzavřeného 2D tvaru (pro kabel MeshCircle). Tvar se zrotuje za pomoci normálového vektoru (s minimalizovanou rotací) a tečny pro daný bod. Dále se mezi vytvořenými vrcholy musí definovat trojúhelníky, které tvoří plochy meshe. Trojúhelníky jsou definovány jako posloupnost čísel indexů vrcholů. Vrcholy trojúhelníků musí být zapsány ve směru hodinových ručiček, aby byli správně zobrazováni. Výpočet meshe může být poměrně náročný (náročnost se odvíjí od rozlišení meshe a jeho délky), proto vytváření meshe během propojování kabelů je zajištěno samostatným vláknem. Výpočet probíhá asynchronicky a mesh kabelu je aktualizován jen pokud vlákno dokončilo výpočet.



■ **Obrázek 5.6** Ukázka směru zápisu Trojúhelníků



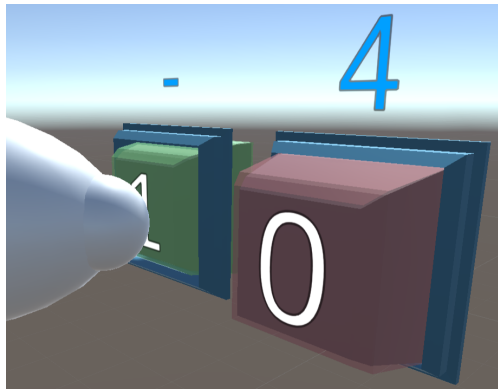
■ **Obrázek 5.7** Ukázka propojování kabelem

5.2.5 Ovládání

Ovládání je zajištěno knihovnou OpenXR, která je propojena s Unity Action Input systémem. To dovoluje jednoduché přiřazování akcí k ovládání. Uchopování objektů je zprostředkováváno vlastním komponentem typu Grabbable. Komponenty při uchopení vrátí instanci objektu, a tak umožňují například uchopit konektor, který vrátí ruce instanci propojovacího kabelu. Podobně je implementovaná mazací hůlka připevněna k levé ruce. Uchopování je zajištěno komponentem GrabController, který je notifikován action input systémem o stavu spouště na ovladači.

Pohyb je dle návrhu zajištěn metodou world pulling. Zvětšování a zmenšování celého objektu XR Rig (hráče) je provedena iluze zmenšování/zvětšování celé úrovně a všech ostatních objektů. Pohyb je implementováno v komponentu GripController, který je notifikován action input systémem o stavu sekundárního tlačítka na ovladači.

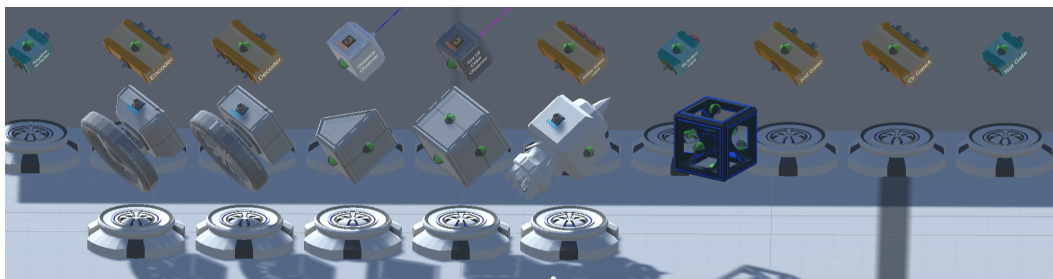
Pro další interakce je využíváno tlačítka, které hráč aktivuje kolizí ruky s tlačítkem. Kolize jsou zajištěny podobně jako u bloků, tj. komponenty Rigidbody a Collider. Tlačítka se skládá ze dvou částí. První slouží jako tělo tlačítka a má Rigidbody s vlastností kinematic. Druhá část má Rigidbody, ale s vypnutou simulací gravitace a navíc má Configurable Joint, který připojen k tělu tlačítka, a tak umožňuje posun tlačítka.



■ Obrázek 5.8 Ukázka mačkání tlačítka

5.2.6 Modelování

K vytvoření všech modelů jsem využil modelovací program Blender. Pro některé modely jsem vytvořil vlastní materiály a nabarvil jsem jimi jednotlivé plochy. Zatímco pro ostatní modely jsem použil volně dostupný balíček prototypových textur. Pro model první úrovně bylo potřeba vytvořit klikatou cestu, té bylo docíleno deformováním meshe podél definované křivky.



■ Obrázek 5.9 Ukázka modelů ze hry

5.3 Testování

V této podkapitole je přestaveno využití testování během vývoje prototypu hry. Stejně jako při vývoji softwaru, tak i u her je potřeba testovat. Testování během vývoje bylo rozděleno na tři části: automatizované testování, manuální a testování hratelných iterací.

5.3.1 Automatizované testování

Často se automatizované testování u her velmi opomíjí, z důvodu rychlého iterativního vývoje a časové náročnosti pro vytváření automatizovaných testů. V této práci pro některé části kódu bylo vhodné vytvořit automatické testy. Testy jsou rozděleny do dvou hlavních skupin EditMode testy a PlayMode testy, toto rozdělení je způsobeno využitým Unity frameworkem pro testování. Vytvořené testy patří převážně mezi integrační. Integrační typ testů testuje více neizolovaných částí kódu. Dalším typem testů jsou Unit testy, ty testují izolovaný kód (např. na úrovni samotných metod).

EditMode testy jsou určeny pro kód, který nepotřebuje běžící scénu k testování. Často jsou to jednoduché datové třídy, nebo implementace algoritmů. V této práci se v EditMode testuje generování Bézierových křivek, intervalové binární vyhledávání a třídy snibble.

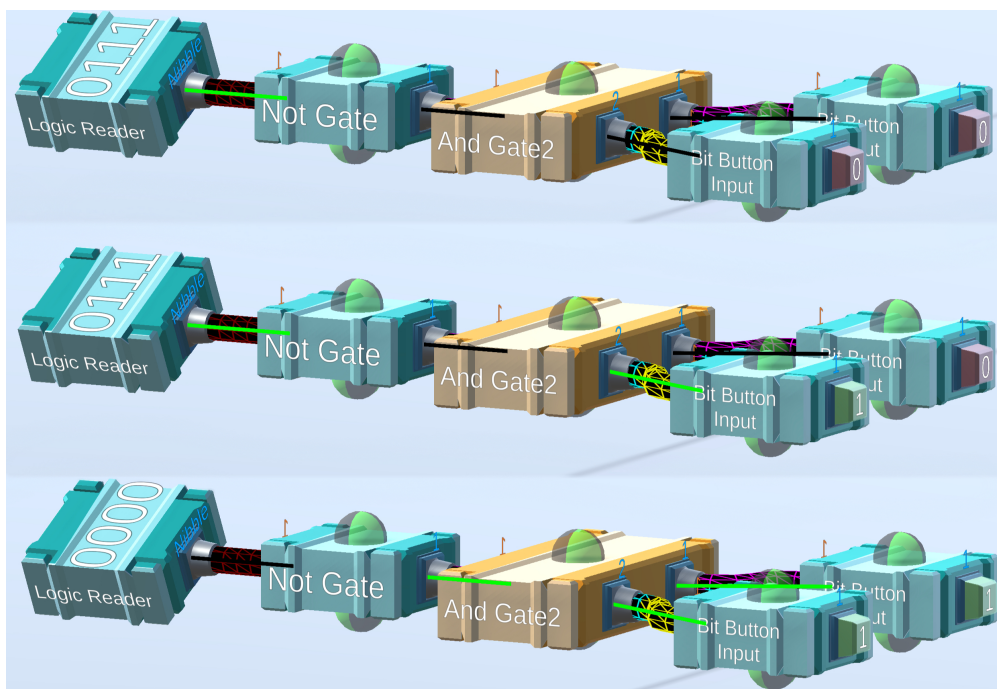
PlayMode testy jsou určeny pro kód, který je závislý na objektech instancovaných v scéně např. na komponentu Transform. Proto jsou testy spuštěny v samostatné scéně určené k testování. V testu je možno instancování potřebných objektů. V Playmode testech testují komunikaci logického systému, protože komponent Receivers vytváří herní objekty konektorů a to je možné pouze v PlayMode testech.

5.3.2 Manuální testování

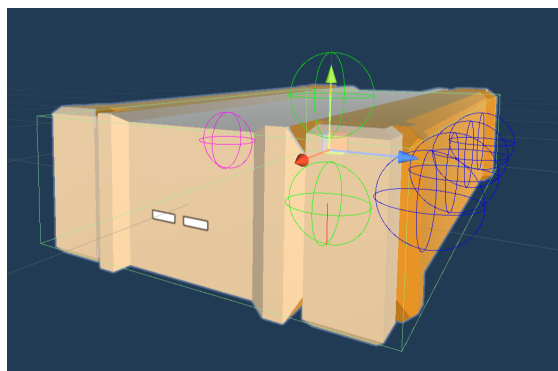
V průběhu vývoje bylo potřeba testovat funkčnost kódu a samotných mechanik. Manuální testování je jedinou možností testování a vyhodnocování mechanik. Probíhá v samotné hře a manuálně jsou testovány nově implementované funkčnosti. Oproti automatizovanému testování, je většinou rychlejší z krátkodobého hlediska a dostáváme také více informací (jestli je mechanika vhodná). Tento typ testování není vhodný pro testování algoritmů a datových struktur, které se moc nemění v průběhu vývoje.

Větší část vývoje jsem s testoval bez VR. Cílem bylo co nejdříve vytvořit iteraci plně testovatelnou ve VR. To znamenalo vytvořit základní funkce editoru: zacvakávání bloků do sebe, základ logického systému a jeho propojování. Testování bez VR bylo dosaženo vytvoření testovací ruky, která se ovládala z Unity editoru. Po základní implementaci další testování probíhalo ve VR z důvodu vyladění herních systémů.

Unity poskytuje možnost kreslení jen v editoru tzv. Gizmos. Ty umožňují znázorňovat důležité informace pro testování a vývoj. Můžou být vykreslovány jen pokud je objekt s Gizmos vybraný (nebo jeho rodič) a nebo vždy. V této práci byli především využity pro vizualizaci signálu mezi jednotlivými propojení součástí, umístění kotev a konektorů. Signál je znázorněn přímkami mezi konektory a různými barvami se určuje typ signálu (černá – signál nuly, zelená – kladný signál, červená – negativní signál).



■ Obrázek 5.10 Ukázka testování obvodů – implementace NAND brány



■ Obrázek 5.11 Ukázka Gizmos pro brány

5.3.3 Interní testování

Během vývoje jsem provedl testování třech větších iterací. V této části budou představeny jednotlivé iterace a poznatky z jejich testování. Testování bylo provedeno s několika uživateli na systému Windows a platformě SteamVR (ekosystém zařízení VR).

5.3.3.1 První iterace

V první iteraci byl implementován pouze stavební systém s možností úchopu dílů a zacvakávání. Dostupné díly byly pouze základní stavební krychle, poháněné kolečko a zatačecí kolečko. Dále byl implementován logický systém s jedním logickým obvodem (násobič mínusem) a blokem pro

detekování vzdálenosti objektu. Vytváření objektů nebylo ještě implementováno, tudíž uživatel měl k dispozici omezený počet dílů. Pohyb byl možný proveden metodou world pulling.

Připomínky a chyby:

- Špatně rozeznatelné vstupy a výstupy.
- Při uchopení bloku propojeného kabelem, kabel nezmizel.
- Pomalý pohyb

5.3.3.2 Druhá iterace

Na základě předchozí iterace byli opraveny chyby a provedeno několik změn: barevně rozlišené konektory a popisky ke konektorům (modrá vstup, oranžová výstup), zrychlení pohybu a opraveno mazání kabelu. Dále v této iteraci přibyla první úroveň, automatické popisky logických bloků, kontrola překrytí pokládaných bloků, změna měřítka, několik nových bloků a obvodů.

Připomínky a chyby:

- Při změně měřítka pocit šilhání.
- Úroveň má moc ostré zatáčky.
- Nelze skládat obvody na sebe (špatná detekce překrývání pokládaných obvodů).
- Detektor objektů má moc krátkou detekční vzdálenost.
- Chybné chování logického obvodu negace snibble.

5.3.3.3 Třetí iterace

Pocit šilhání při změně měřítka, lze přisoudit optickému klamu, který pravděpodobně vzniká tím že změna měřítka světa je simulována změnou měřítka samotného hráče (XR rigu). Změnou měřítka hráče se zmenšuje vzdálenost mezi kamerami pro každé oko a ostatních objektů hráče. Tato změna vzdálenosti je možný důvod pocitu šilhání, protože objekty zůstávají na stejném místě a vzdálenost mezi objekty a hráčem zůstává stejná. Problém pocitu šilhání jsem z větší části omezil tak, že posouvám hráčem dopředu a dozadu dle změny měřítka. Další problémy z minulé iterace byli opraveny. V poslední iteraci přibyla další úroveň a nový vstupní logický blok, který rozpoznává tmavost objektu před ním. Dále přibilo: texturování objektů, statické objekty pro získávání dílů, nové díly, časomíra, cíl a manuální generátor signálu s tlačítky.

Připomínky a chyby:

- Při změně mapy některá tlačítka přestanou fungovat.
- Raketový motor aktivuje pouze částicové efekty.
- Při přejezdu mostu v nové úrovni se občas vozítko zasekne.
- Chybí tutoriál.

Kromě chybějícího tutoriálu byli připomínky a chyby opraveny.

Optimální herní strategie

V této kapitole je nejdříve krátce představena teorie her a payoff matice. Dále je sestavena payoff matice s možnými strategiemi hraní hry.

6.1 Teorie her

Teorie her pomáhá porozumět interakci rozhodujících se aktérů. Hra (v teorii her) je v podstatě kompetitivní aktivita kde hráči soutěží mezi sebou dle daných pravidel. Teorie her dokáže popsat širokou škálu aktivit, může zahrnovat klasické hry, politiku, ekonomiku... Dělí se na dvě hlavní části kooperativní a kompetitivní. Kooperativní se soustředí na rozhodování skupin hráčů (aktérů), tyto skupiny jsou označovány jako koalice. Kompetitivní se soustředí na rozhodování jednotlivců v jejich prospěch. V této práci se zaměříme na kompetitivní hry. [24]

Zřejmě nejznámější použití teorie her je věžňovo dilema. Zkrácený příběh k dilematu je následující. Policie chytila dva zločince a má dostatek důkazů pro jejich obvinění z přestupku. Policie ví, že spáchali trestný čin, ale nemá pro to jednoznačné důkazy, proto zločince rozdělila. Zločinci se mohou buď přiznat, nebo mlčet. Vznikají tak tři (čtyři, jedna je zdvojená) možnosti:

- Oba mlčí – odnětí svobody na dva měsíce
- Jeden se přizná – ten co se přizná je volný a druhý má trest na 12 měsíců
- Oba se přiznají – odnětí svobody na osm měsíců

Zločinci nemohou nijak sdílet informace, proto si rovněž nemohou být jisti, jestli jeden druhého nezdáří. Pro lepší porozumění problému je dobré sestavit payoff matici. Ta reprezentuje kombinace rozhodnutí hráčů (věžňů) a jejich zisk/ztrátu pro dané rozhodnutí. Levá dolní strana ohodnocení patří hráči 1 a pravá horní strana hráči 2. Matice pro věžňovo dilema vypadá následovně.

■ **Tabulka 6.1** Payoff matice pro věžňovo dilema

		Hráč 2	
		Mlčení	Přiznání
Hráč 1	Mlčení	-2	0
	Přiznání	-12	-8

Z matice můžeme určit, jaká je správná volba akce pro hráče. Předpokládejme, že hráč 1 ví, že hráč 2 bude mlčet, pak je pro hráče 1 nejlepší se přiznat, protože platí $0 > -2$. Dále pokud se hráč 2 přizná, tak pro hráče 1 je opět nejlepší se přiznat. Pro hráče 1 je tedy optimální strategií přiznat se, a to nezávisle na rozhodnutí hráče 2. Říkáme, že strategie zrady striktně dominuje strategii spolupráce i přesto, že má spolupráce v součtu lepší výsledek [24]. Důvodem dominance je, že hráči jsou racionální a snaží se maximalizovat svůj zisk.

Nashova rovnováha je stav, kdy jednostranná změna strategie není pro hráče výhodná [24]. Pro hru lov na jelena nalezneme Nashovy rovnováhy. Lovci (hráči) si mohou vybrat lovit jelena nebo zajíce, jelena je možno ulovit jen pokud budou všichni lovci spolupracovat. Ulovit zajíce je jednodušší a každý lovec, který se rozhodne lovit zajíce, jednoho uloví. Lovci jsou motivováni lovit jelena protože část z jelena poskytuje větší zisk než zajíc. Pro možnost jednoduchého vytvoření payoff matice se předpokládá hra se dvěma lovci.

■ **Tabulka 6.2** Payoff matice pro lov jelena

		Hráč 2	
		Jelen	Zajíc
Hráč 1	Jelen	2	1
	Zajíc	0	1

Hra má dvě rovnováhy. První je spolupráce při lovení jelena a druhá je lov zajíců. Jestliže jeden hráč chce lovit jelena, druhému hráči se vyplatí jít lovit jelena. To samé platí i pro lov zajíců. Jeden hráč nemůže ulovit jelena samostatně, proto je nucen jít také lovit zajíce. Věžňovo dilema má pouze jednu rovnováhu, protože hráči mohou oproti mlčení získat více, pokud se přiznají.

6.2 Optimální herní strategie

Sestavení payoff matice pro hru s jedním hráčem trivializuje problémy, kterými se teorie her zabývá. Vzniklá matice bude mít pouze jeden sloupec a v zásadě řeší pouze optimalizační problém, protože hráč soutěží pouze o nejrychlejší čas.

Porovnáme dvě herní strategie vozítek. První vozítko se zatáčejícími kolečky vpředu a druhé se zatáčejícími kolečky v zadní části. První vozítko dokázalo využít maximální rychlosti (signál o síle 7), ale při rychlostech vyšších než čtyři docházelo k malé srážce se stěnou, i přes srážku nejvyšší rychlost byla nejlepší. Druhé vozítko dosáhlo jen čtvrté rychlosti s malou srážkou, při

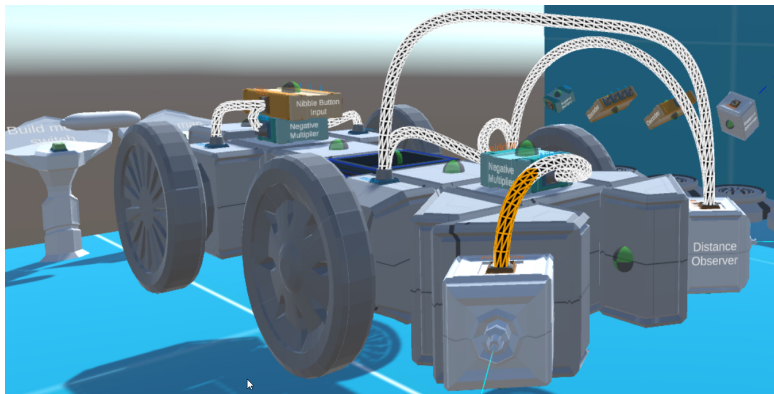
vyšších rychlostech docházelo k úplnému zaseknutí o stěnu. Výsledné časy jsou zaneseny do payoff matice.

■ **Tabulka 6.3** Payoff matice různých strategií vozítek

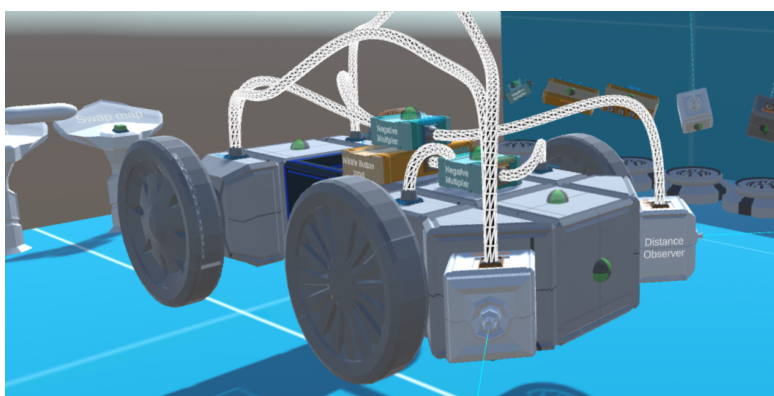
Hráč 2		
Přední rychlost 4	Přední rychlost 7	Zadní rychlost 4
02:37,9	01:58,9	03:06,4

Z matice můžeme vidět, že zatáčení předními kolečky je lepší strategie i přesto, že vozítko využilo více bloků a bylo tak těžší. Vozítko se zadním zatáčením bylo zpomaleno srážkou s okrajem dráhy, ale také delší cestou, které bylo způsobné pomalejším zatáčením vozítka.

Dále jsem také testoval různé strategie sledování čáry v druhé úrovni. U tohoto vozítka byla limitující oscilace sledování čáry. Je potřeba balancovat rychlost s úhlem zatáčení koleček, tak aby vozítko co nejplynuleji sledovalo čáru. Oscilace vozítka často způsobovala, že vozítko přestalo sledovat čáru.



■ **Obrázek 6.1** Ukázka vozítka se zatáčením vpředu



■ **Obrázek 6.2** Ukázka vozítka se zatáčením vzadu

Finanční plán

Finančním plánem se snažíme předvídat příjmy a výdaje. Často bývá jako hlavní součást podnikatelského plánu, v tomto případě pomůže navrhnout proveditelný rozvoj hry. V plánu musíme zahrnout všechny prostředky spojené s podnikáním např.: provozní náklady, prostředky na založení, pořízení dlouhodobého majetku... [32]

Prvně je potřeba sestavit zahajovací rozvahu, ta poskytuje informace o počáteční finanční a majetkové situaci firmy. Má dvě části aktiva a pasiva, jednotlivé položky musí být vyčísleny. Stejně jako v podvojném účetnictví aktiva a pasiva se musí rovnat. Aktiva jsou majetek, který přináší nějaký budoucí ekonomický prospěch. Pasiva jsou zdroje krytí tohoto majetku. [25]

Dále je potřeba sestavit odhady výkazy zisků a ztrát. Na počátku se předpokládá převážně se ztrátami, do té doby je potřeba všechny náklady hradit z vlastních, nebo cizích zdrojů. Je potřeba zohlednit všechny aspekty zisků a výdajů (mzdy, nákup materiálu, provozní náklady...). Při odhadování je vhodné stanovit tři varianty odhadů: pesimistický, realistický a optimistický. Tyto varianty pak pomáhají pochopit rizika a proveditelnost samotného plánu.

7.1 Zahajovací rozvaha

Na počátku společnosti bude hra ve finálním vývoji první verze. Bude potřeba nakoupit nové platformy HMD VR pro testování funkčnosti a potřebného výkonu. Celková cena VR HMD vychází přibližně na 55 000 Kč (Valve Index – 27 000 Kč, HP Reverb – 15 000 Kč, Oculus Quest 2 – 13 000 Kč). Dále bude potřeba pořídit nový počítač a periferie počítač vychází na 50 000 Kč a periferie na 15 000 Kč. Bude potřeba zakoupit softwarové licence za 9 000 Kč (Windows pro, verzovací systém Bitbucket a WinRAR). Jako základní kapitál vložím do firmy 200 000 Kč z vlastních zdrojů. Společnost si bude muset vzít podnikatelský úvěr ve výši 200 000 Kč na čtyři roky s úrokem 5,9 % s měsíční splátkou 5 000 Kč.

■ **Tabulka 7.1** Zahajovací rozvaha

Aktiva		Pasiva	
Věc	Částka	Věc	Částka
VR HMD	55 000 Kč	Základní kapitál	200 000 Kč
Počítač a periferie	65 000 Kč	Úvěr	200 000 Kč
Softwarové licence	9 000 Kč		
Bankovní účet	271 000 Kč		
Součet aktiv	400 000 Kč	Součet pasiv	400 000 Kč

7.2 Odhad výdajů a zisků

Počítá se s minimálními náklady na vývoj, protože část investice bude časová z mé strany. Jsem schopen poskytnout přibližně 3 MD (člověkodnů – osm hodin práce) týdně díky současné formě zaměstnání. V prvním roce se počítá s vydáním první verze hry s předběžným přístupem. Jedná se o častý krok vývojářů her pro získání dalšího kapitálu a zpětné vazby od uživatelů. Hry s předběžným přístupem bývají prodávány se slevou jako bonus pro první uživatele, budeme počítat se slevou 20 % oproti cílové ceně. První tři měsíce se bude pokračovat na dokončování vývoje. Pro dokončení hry bude potřeba najmout profesionálního grafika na 60 MD práce (přibližně 3 kalendářní měsíce). Při 300 Kč za hodinu je cena práce pro 3D grafika 144 000 Kč. Během čtvrtého měsíce bude vydána verze hry s předběžným přístupem na platformě Steam a Oculus Store, obě platformy si automaticky z prodeje berou přibližně 30% ceny. Pro platformu Steam je navíc potřeba zaplatit jednorázový poplatek 3 000 Kč. V prvním měsíci vydání hry bude provedena malá reklamní kampaň platformě YouTube a Facebook reklamy s rozpočtem 40 000 Kč. Reklamy pro ty to platformy se platí za kliknutí (přibližně 2-10 Kč za klik).

Dále v průběhu prvního roku a první poloviny druhého bude znovu potřeba najmout grafika na dalších 60 MD. V půlce druhého roku se počítá s vydáním konečné verze hry (cena hry bude 500 Kč) a větší marketingovou kampaní s rozpočtem 150 000 Kč. Druhý a třetí rok je také možno předpokládat potřeba zakoupení nově vydaných VR HMD pro testování, pořizovací cenu těchto HMD stanovíme na 30 000 Kč pro druhý a 40 000 Kč pro třetí rok. Další roky je potřeba obnovit licence Bitbucket ta na rok stojí 1 700 Kč. Třetí rok počítáme s rozpočtem na reklamu 100 000 Kč. Cenu her jsme dle průzkumu trhu určili na 500 Kč (350 Kč po poplatku platformem) zlevněnou verzi na 400 Kč (280 Kč po poplatku platformem). Navíc je potřeba zaplatit 60 000 Kč ročně na úrocích pro podnikatelský úvěr a web hosting 1 500 Kč měsíčně. Následující odhady jsou provedeny pomocí různých odhadů oslovení uživatelů.

7.2.1 Realistický odhad

Dle realistického odhadu počtu cílových uživatelů budeme předpokládat, že 1 % z nich si zakoupí hru v prvním měsíci prodeje. Odhaduji pouze jedno procento, protože oslovení her s předběžným přístupem je nižší. Celkem v prvním měsíci prodeje bude 750 prodaných kusů za celkovou cenu 210 000 Kč (již odečten 30 % poplatek). Další tři měsíce předpokládám růst celkové skupiny uživatelů 5 % za měsíc další měsíce do vydání plné verze předpokládám 1,5 %, tedy celkem za první rok 935 prodaných her za celkovou cenu 261 800 Kč. Růst 1,5 % za měsíc jsem stanovil dle předpokládaného růstu celkového trhu. Druhý rok před vydáním konečné verze se předpokládá prodej 87 her (24 360 Kč) a při vydání konečné verze je odhadováno oslovení 4 % procent cílového trhu a stejného růstu jako při předchozím vydání (první tři měsíce 5 % dále 1,5 %).

Druhý rok z prodeje her získáme 1 345 260 Kč (3 861 kopií her). Ve třetím roce počítáme s konstantním růstem uživatelů 1,5 % za měsíc. Celkem za třetí rok prodáme 866 her za celkovou cenu 303 100 Kč.

■ **Tabulka 7.2** Realistický odhad výdajů a zisků

	1. rok	2. rok	3. rok
Výnosy	261 800 Kč	1 345 260 Kč	303 100 Kč
Prodej	261 800 Kč	1 345 260 Kč	303 100 Kč
Náklady	521 500 Kč	387 200 Kč	203 200 Kč
Reklama	40 000 Kč	150 000 Kč	100 000 Kč
Grafik	288 000 Kč	144 000 Kč	–
Ostatní	193 500 Kč	93 200 Kč	103 200 Kč
Zisk/ztráta	-259 700 Kč	958 060 Kč	99 900 Kč

7.2.2 Optimistický odhad

U optimistického odhadu budeme předpokládat optimistickou velikost cílového trhu (100 000 potenciálních uživatelů) a rychlejší růst skupiny uživatelů v pozdějších stádiích namísto 1,5 % budeme počítat s 2 % za měsíc. Ostatní hodnoty se budou dopočítávat stejně jako u realistického odhadu. Celkem v prvním měsíci prodeje oslovíme 1 % cílového trhu, tedy 1 000 uživatelů. S optimistickým růstem skupiny uživatelů za první rok na prodejích získáme 375 840 Kč (1278 prodaných her). Druhý rok s vydáním nové verze z prodeje získáme 1 833 930 Kč (161 zlevněných kopií a 5 111 za plnou cenou). Třetí rok z prodeje získáme 569 100 Kč.

■ **Tabulka 7.3** Optimistický odhad výdajů a zisků

	1. rok	2. rok	3. rok
Výnosy	375 840 Kč	1 833 930 Kč	569 100 Kč
Prodej	375 840 Kč	1 833 930 Kč	569 100 Kč
Náklady	521 500 Kč	387 200 Kč	203 200 Kč
Reklama	40 000 Kč	150 000 Kč	100 000 Kč
Grafik	288 000 Kč	144 000 Kč	–
Ostatní	193 500 Kč	93 200 Kč	103 200 Kč
Zisk/ztráta	-163 660 Kč	1 446 730 Kč	365 900 Kč

7.2.3 Pesimistický odhad

U pesimistického odhadu budeme předpokládat pesimistickou velikost cílového trhu (50 000 potenciálních uživatelů) a pomalejší růst skupiny uživatelů v pozdějších stádiích namísto 1,5 % budeme počítat s 1 % za měsíc. Z odhadu můžeme vidět, že třetí rok jsme snížili rozpočet pro reklamu a nenakoupili nové technologie. To způsobuje pomalejší růst skupiny uživatelů. V tomto případě bude dostatek výnosů na splácení úvěru, ale celkové zisky budou relativně malé, tudíž budou ztráty časové z mé strany. Časové ztráty se dají převést na peněžní, během prvních dvou let vývoje bych strávil přibližně 288 MD. Při hodinové sazbě 300 Kč by převedená časová ztráta měla hodnotu 691 200 Kč. Z toho můžeme usoudit, že v tomto případě se vývoj hry nevyplatí.

	1. rok	2. rok	3. rok
Výnosy	170 335 Kč	877 660 Kč	127 750 Kč
Prodej	170 335 Kč	877 660 Kč	127 750 Kč
Náklady	521 500 Kč	387 200 Kč	113 200 Kč
Reklama	40 000 Kč	150 000 Kč	50 000 Kč
Grafik	288 000 Kč	144 000 Kč	–
Ostatní	193 500 Kč	93 200 Kč	63 200 Kč
Zisk/ztráta	-351 165 Kč	490 460 Kč	14 550 Kč

■ **Tabulka 7.4** Pesimistický odhad výdajů a zisků

Diskuze a závěr

Cílem této práce bylo pro počáteční koncept logické VR hry provést analýzu desktopových logických her, průzkum trhu a nalézt cílový trh. Dalším cílem bylo na základě předchozí analýzy navrhnout, implementovat a otestovat prototyp nové logické hry a vytvořit finanční plán pro další rozvoj hry.

Výsledkem práce je funkční prototyp logické VR hry, kde cílem je postavit autonomní vozítko, které má se má co nejrychleji dostat do cíle. Prototyp nabízí hráči dvě úrovně. První je jednodušší vozítko musí projet klikatou cestou. Druhá je složitější, inspirována tratěmi pro vozítka tzv. autonomus line follower. Vozítko musí sledovat černou čáru, aby se dostalo do cíle, ale po cestě se musí ještě zastavit před padacím mostem. Autonomizace je dosažena pomocí logického systému využívající polovičního bytu se znaménkem pro komunikaci mezi logickými bloky.

Hra je inspirována již existujícími tituly, dle kterých byl navržen herní cyklus. Vývoj hry je díky rozsáhlosti ve svých začátcích. Koncept umožnil otestování základního herního cyklu a vyhodnocení jeho nedostatků. Je potřeba dále hru rozšířit o mnoho důležitých součástí jako např.: menu, tutoriál, tvorba vlastních obvodu, ukládání... Dále je potřeba otestovat koncept na jiných zařízeních VR. Otestování hry na standalone VR za účelem nalezení nedostatečně optimalizovaných částí hry je důležité, protože standalone VR představuje značnou část trhu.

Prototyp i přes kompromisy při vývoji poskytuje jednoduchý a intuitivní editor. Vozítka v editoru lze vytvářet poměrně rychle. Získávání dílů pro stavbu je zajištěno statickými objekty v úrovni, které díly při odebrání vytvoří nový díl. Díky možnosti měřítka je získávání dílů poměrně rychlé, ale pokud se výběr dílů více rozšíří, tak výběr začne být nepřehledný. V návrhu byla navržena možnost připevněného menu k ruce. Tato možnost by zpřehlednila výběr dílů a odstranila potřebu změny měřítka pro získání dílů. Proto by bylo vhodné tuto metodu získávání dílů implementovat.

Implementace logického systému by mohla být dále vylepšena, za cílem snížení nároků na hardware. Simulace průběhu logických obvodů lze provést odděleně od logických obvodů a pouze vstupní/výstupní díly by komunikovali se systémem. Systém by tak reagoval jen při změně vstupu a logické obvody by logický systém jen reprezentovali. Bylo by potřeba provést další testování a návrh celého systému, jestli by toto řešení bylo vhodné, potřebné a proveditelné. Dalším vylepšením systému by mohla být možnost zpracování více snibble vstupů/výstupů zároveň, každý logický díl má limit jednoho snibble konektoru pro vstup a výstup. Systém by mohl být ještě rozšířen o klopné obvody.

Průzkum trhu a finanční plán mohou posloužit jako základ pro potencionální podnikatelský plán. V rámci toho by bylo potřeba provést primární průzkum trhu, za účelem zjištění očekávání zákazníků, zároveň by bylo vhodné vyzkoušet prototyp na větší skupině lidí. Z finančního plánu je zřejmé, že další vývoj hry je finančně proveditelný, ale není bez rizika.

Díky testování prototypu a vytvoření finančního plánu, bych rád ve vývoji hry pokračoval. Prototyp hry je zábavný a má velký potenciál. Hra může sloužit jako představení návrhu jednoduchých logických obvodů.

Bibliografie

1. Abrupt rotation changes in frames resulting from perpendicular vector construction. In: *Medium* [online]. A Medium Corporation, 2021 [cit. 2022-04-24]. Dostupné z: <https://medium.com/intuition/lockdown-geometry-rotation-minimizing-frames-ff373d2f355b>.
2. BÉZIER, Pierre E; SIOUSSIOU, Salah. Semi-automatic system for defining free-form curves and surfaces. *Computer-Aided Design*. 1983, roč. 15, č. 2, s. 65–72. ISSN 0010-4485. Dostupné z DOI: [https://doi.org/10.1016/0010-4485\(83\)90170-7](https://doi.org/10.1016/0010-4485(83)90170-7).
3. *Blender* [online]. Blender Foundation, 1998. [cit. 2022-04-25]. Dostupné z: <https://www.blender.org/>.
4. BOAS, YAGV. Overview of virtual reality technologies. In: *Proceedings of the Interactive Multimedia Conference*. University of Southampton, 2013.
5. Consumer VR headset active installed base and content revenue, 2017-2026. In: *Game Developer* [online]. Informa PLC, 2021 [cit. 2022-04-17]. Dostupné z: <https://www.gamedeveloper.com/blogs/omdia-research-reveals-12-5-million-consumer-vr-headsets-sold-in-2021-with-content-spend-exceeding-2bn>.
6. Creating Prefabs. In: *Unity Documentation* [online]. v2020.3. vyd. Unity Technologies, 2022 [cit. 2022-04-25]. Dostupné z: <https://docs.unity3d.com/2020.3/Documentation/Manual/CreatingPrefabs.html>.
7. GIGANTE, Michael A. Virtual Reality: Definitions, History and Applications. In: EARN-SHAW, R.A.; GIGANTE, M.A.; JONES, H. (ed.). *Virtual Reality Systems*. London: Academic Press, 1993, s. 4–14. ISBN 978-0-12-227748-1. Dostupné z DOI: <https://doi.org/10.1016/B978-0-12-227748-1.50009-3>.
8. GRAUBALLE, Morten. Forget about Gamification – It is All about the Gameplay Loop! In: *Developer Nation* [online]. SlashData Ltd, 2015 [cit. 2022-04-12]. Dostupné z: <https://www.developernation.net/blog/forget-about-gamification-it-is-all-about-the-gameplay-loop>.
9. HAGEN, Hans. Bezier-curves with curvature and torsion continuity. *The Rocky Mountain Journal of Mathematics*. 1986, s. 629–638. Dostupné také z: <https://www.jstor.org/stable/44237020>.
10. HAMILTON, Ian. Exclusive: Here's How Fantastic Contraption VR Devs Generated More Than \$1 Million. In: *UploadVR* [online]. UVR Media LLC, 2017 [cit. 2022-04-17]. Dostupné z: <https://uploadvr.com/fantastic-contraption-fuels-devs-next-projects-generates-1-million/>.
11. Introduction to components. In: *Unity Documentation* [online]. v2020.3. vyd. Unity Technologies, 2022 [cit. 2022-04-25]. Dostupné z: <https://docs.unity3d.com/2020.3/Documentation/Manual/Components.html>.

12. JIJASHVILI, George. Omdia research reveals 12.5\$ million consumer VR headsets sold in 2021 with content spend exceeding \$2bn. In: *Game Developer* [online]. Informa PLC, 2021 [cit. 2022-04-17]. Dostupné z: <https://www.gamedeveloper.com/blogs/omdia-research-reveals-12-5-million-consumer-vr-headsets-sold-in-2021-with-content-spend-exceeding-2bn>.
13. Joints. In: *Unity Documentation* [online]. v2020.3. vyd. Unity Technologies, 2022 [cit. 2022-04-25]. Dostupné z: <https://docs.unity3d.com/2020.3/Documentation/Manual/Joints.html>.
14. KLOK, Fopke. Two moving coordinate frames for sweeping along a 3D trajectory. *Computer Aided Geometric Design*. 1986, roč. 3, č. 3, s. 217–229. ISSN 0167-8396. Dostupné z DOI: [https://doi.org/10.1016/0167-8396\(86\)90039-7](https://doi.org/10.1016/0167-8396(86)90039-7).
15. MANJREKAR, Siddhesh; SANDILYA, Shubhrika; BHOSALE, Deesha; KANCHI, Sravanthi; PITKAR, Adwait; GONDHALEKAR, Mayur. CAVE: an emerging immersive technology—a review. In: *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*. IEEE, 2014, s. 131–136. Dostupné z DOI: 10.1109/UKSim.2014.20.
16. MIROSLAV, Koprla. Marketingový průzkum trhu jako nástroj zvyšování konkurenceschopnosti. In: *Dny práva – 2009 – Days of Law: the Conference Proceedings*. Brno: Masaryk University, 2009, s. 881–2892. ISBN 978-80-210-4990-1.
17. MORA-CANTALLOPS, Marçal. Transhistorical perspective of the puzzle video game genre. In: *Proceedings of the 13th International Conference on the Foundations of Digital Games* [online]. Malmö, Sweden: Association for Computing Machinery, 2018, s. 1–9. ISBN 9781450365710. Dostupné také z: <https://doi.org/10.1145/3235765.3235768>.
18. NORTHWAY GAMES. *Fantastic Contraption* [soft.]. v2.02. vyd. 2016. [cit. 2022-04-14]. Dostupné z: https://store.steampowered.com/app/386690/Fantastic_Contraption/.
19. Obchod steam. In: *Steam* [online]. Valve Corp., 2022 [cit. 2022-04-17]. Dostupné z: <https://store.steampowered.com/search/?tags=21978%2C1664&vrssupport=402>.
20. Oculus Rift. In: *VR bound* [online]. VR Bound LTD, 2022 [cit. 2022-04-09]. Dostupné z: <https://www.vrbound.com/headsets/oculus/rift>.
21. Oculus Rift: Step Into the Game. In: *Kickstarter* [online]. Kickstarter, PBC, 2016 [cit. 2022-04-08]. Dostupné z: <https://www.kickstarter.com/projects/1523379957/oculus-rift-step-into-the-game/posts>.
22. OLMOS RAYA, Elena; FERREIRA, Janaina; SOLER-DOMINGUEZ, Jose; CONTERO, Manuel; ALCANIZ RAYA, Mariano; OLMOS, E; CAVALCANTI, Janaina. Mobile Virtual Reality: A Promising Technology to Change the Way We Learn and Teach. In: *Mobile and Ubiquitous Learning: An International Handbook* [online]. Singapore: Springer Singapore, 2018, s. 96–97. ISBN 978-981-10-6144-8. Dostupné také z: https://doi.org/10.1007/978-981-10-6144-8_6.
23. *OpenXR* [online]. The Khronos, 2022. [cit. 2022-04-25]. Dostupné z: <https://www.khronos.org/openxr/>.
24. OSBORNE, Martin J. *An introduction to game theory*. New York: Oxford University Press, 2003. ISBN 9780195128956.
25. POKORNÁ, Lenka. Jaké informace vyčteme z rozvahy? In: *Podnikatel* [online]. Internet Info, 2021 [cit. 2022-05-01]. Dostupné z: <https://www.podnikatel.cz/clanky/jake-informace-vycteme-z-rozvahy/>.
26. PRIVATE DIVISION. *Kerbal Space Program* [soft.]. v1.12.3. vyd. 2015. [cit. 2022-04-13]. Dostupné z: https://store.steampowered.com/app/220200/Kerbal_Space_Program.

27. PropertyDrawer. In: *Unity Documentation* [online]. v2020.3. vyd. Unity Technologies, 2022 [cit. 2022-04-25]. Dostupné z: <https://docs.unity3d.com/ScriptReference/PropertyDrawer.html>.
28. Rigidbody. In: *Unity Documentation* [online]. v2020.3. vyd. Unity Technologies, 2022 [cit. 2022-04-25]. Dostupné z: <https://docs.unity3d.com/2020.3/Documentation/Manual/class-Rigidbody.html>.
29. RMFs on curve, notice smooth rotational transition between frames and axis alignment. In: *Medium* [online]. A Medium Corporation, 2021 [cit. 2022-04-24]. Dostupné z: <https://medium.com/intuition/lockdown-geometry-rotation-minimizing-frames-ff373d2f355b>.
30. SCHMITT, Yuko S; HOFFMAN, Hunter G; BLOUGH, David K; PATTERSON, David R; JENSEN, Mark P; SOLTANI, Maryam; CARROUGHER, Gretchen J; NAKAMURA, Dana; SHARAR, Sam R. A randomized, controlled trial of immersive virtual reality analgesia, during physical therapy for pediatric burns. *Burns*. 2011, roč. 37, č. 1, s. 61–68. ISSN 0305-4179.
31. SPIDERLING STUDIOS. *Besiege* [soft.]. v1.10. vyd. 2020. [cit. 2022-04-13]. Dostupné z: <https://store.steampowered.com/app/346010/Besiege/>.
32. SRPOVÁ, Jitka. *Podnikatelský plán a strategie*. Praha: Grada, 2011. ISBN 978-80-247-4103-1.
33. The global virtual reality market is projected to grow from \$6.30 billion in 2021 to \$84.09 billion in 2028 at a CAGR of 44.8% in the forecast period, 2021-2028... In: *Fortune business insights* [online]. Fortune Business Insights Pvt. Ltd., 2021 [cit. 2022-04-17]. Dostupné z: <https://www.fortunebusinessinsights.com/industry-reports/virtual-reality-market-101378>.
34. The Inspector window. In: *Unity Documentation* [online]. v2020.3. vyd. Unity Technologies, 2022 [cit. 2022-04-25]. Dostupné z: <https://docs.unity3d.com/2020.3/Documentation/Manual/UsingTheInspector.html>.
35. Unit Testing. In: *Unity Documentation* [online]. v2020.3. vyd. Unity Technologies, 2022 [cit. 2022-04-25]. Dostupné z: <https://docs.unity3d.com/2020.3/Documentation/Manual/testing-editor-test-runner.html>.
36. *Unity* [soft.]. V2020.3.31f1. Unity Technologies, 2004. [cit. 2022-04-25]. Dostupné z: <https://unity.com/>.
37. ZIMMERMANN, Peter. Virtual reality aided design. A survey of the use of VR in automotive industry. In: *Product Engineering: Tools and Methods Based on Virtual Reality*. Dordrecht: Springer Netherlands, 2008, s. 277–296. ISBN 978-1-4020-8200-9. Dostupné také z: https://doi.org/10.1007/978-1-4020-8200-9_13.

Obsah přiloženého média

readme.txt	stručný popis obsahu média
exe	adresář se spustitelnou formou implementace
├─ manual.txt	informace ke spuštění a ovládání aplikace
├─ AutonomusVehicleBuilder.rar	zkomprimovaná verze spustitelné aplikace
media	
├─ trailer.mp4	video ukázka aplikace
src	
├─ thesis	zdrojová forma práce ve formátu L ^A T _E X
text	text práce
├─ thesis.pdf	text práce ve formátu PDF