

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informatiky a kvantitativních metod

**Rozšířená realita v prostorové scéně
s využitím mobilních zařízení**
Diplomová práce

Autor: Milan Košťák
Studijní obor: Aplikovaná informatika

Vedoucí práce: Ing. Bruno Ježek, Ph.D.

Hradec Králové

leden 2018

Prohlášení

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 30. 1. 2018

Milan Košťák

Poděkování

Rád bych poděkoval vedoucímu práce Ing. Bruno Ježkovi, Ph.D. za pomoc, cenné rady a podněty při zpracování mé diplomové práce.

Anotace

Diplomová práce zkoumá možnosti využití rozšířené reality v kombinaci s mobilními zařízeními v prostředí webových prohlížečů. Soustředuje se na teoretický popis rozšířené reality a její možné aplikace. Dále se věnuje speciálním zařízením pro rozšířenou realitu a podrobně vysvětluje způsob, jakým mohou být webové prohlížeče používány. Zaměřuje se zejména na metody získávání dat ze senzorů a obrazu kamery a jejich následné aplikace v rámci WebGL.

Praktickým cílem práce je navrhnout a implementovat soubor vzorových úloh využívajících různých senzorů, které mobilní zařízení obsahují. Základem všech úloh je získání údajů ze senzorů a zpracování obrazu z kamery. Každá ze tří vytvořených aplikací představuje jedno ze široké škály různorodých využití získaných dat.

Annotation

Title: Augmented reality in the spatial scene using mobile devices

The diploma thesis explores possibilities of usage of augmented reality in combination with mobile devices inside web browsers environment. It focuses on theoretical description of augmented reality and its possible applications. It also discusses special devices for augmented reality and explains in detail how web browsers are used. It concentrates mainly on methods of obtaining data from the sensors and the camera image and their subsequent application within WebGL.

The practical goal of the thesis is to design and implement a set of sample applications that use a variety of sensors which mobile devices contain. The basis of all samples is to obtain data from the sensors and process the camera image. Each of the three created applications represents one of a wide variety of usage of obtained data.

Obsah

1. Úvod.....	1
2. Rozšířená realita.....	3
2.1. Historie.....	3
2.2. Principy augmentace	5
2.2.1. Video-see-through augmentace	5
2.2.2. Optical-see-through augmentace	6
2.3. Druhy augmentace.....	7
2.4. Mobilní rozšířená realita	7
2.5. Porovnání s virtuální realitou	9
2.6. Rozšířená virtualita	11
2.7. Aplikace rozšířené reality	12
2.7.1. Zdravotnictví	12
2.7.2. Turismus	13
2.7.3. Vzdělávání	15
2.7.4. Nakupování.....	15
2.7.5. Televizní přenosy a reklamy	16
2.7.6. Počítačové hry.....	17
2.7.7. Budoucí aplikace	21
3. Hardware pro rozšířenou realitu.....	22
3.1. Senzory.....	23
3.2. Systémy k určení pozice	24
3.3. Specializovaná zařízení.....	24
3.3.1. Google Glass	25
3.3.2. Microsoft HoloLens	25
3.3.3. Lenovo New Glass C220	26
3.3.4. ZapBox	26

3.3.5.	Budoucnost zařízení pro rozšířenou realitu	26
4.	Využití webového prohlížeče	28
4.1.	JavaScript.....	28
4.2.	WebGL.....	29
4.2.1.	WebGL 1	30
4.2.2.	WebGL 2	31
4.2.3.	Budoucnost akcelerované grafiky na webu	31
4.3.	Údaje ze senzorů.....	32
4.3.1.	Orientace zařízení.....	33
4.3.2.	Pohyb zařízení	35
4.4.	Vstup z kamery.....	36
4.5.	Virtuální a rozšířená realita v prohlížeči.....	38
4.5.1.	WebVR	38
4.5.2.	WebXR.....	40
5.	Návrh a implementace ukázkových úloh	42
5.1.	Popis cíle.....	42
5.2.	Detekce posunutí.....	42
5.3.	Detekce orientace.....	46
5.4.	Způsob komunikace	47
5.5.	Struktura úloh.....	48
5.5.1.	Kontrolér.....	48
5.5.2.	Přijímač.....	50
5.5.3.	Server	50
5.5.4.	Komunikace mezi kontrolérem a přijímačem.....	51
5.6.	Úlohy	51
5.6.1.	Abeceda.....	51
5.6.2.	Pong.....	52

5.6.3.	Bludiště.....	53
5.7.	Pomocné knihovny	54
5.7.1.	transforms3d.js	54
5.7.2.	webgl-utils.js.....	55
5.7.3.	math2.js	55
5.7.4.	WS.js.....	55
5.7.5.	sender.js	57
5.8.	Dostupnost ukázek	57
6.	Výsledky a hodnocení.....	58
6.1.	Dostupnost rozšířené reality	58
6.2.	Vhodnost použití JavaScriptu	58
6.3.	Rychlost detekce markeru z obrazu.....	59
6.4.	Dostupnost senzorů	62
6.5.	Rychlost odezvy	62
7.	Závěr.....	65
8.	Reference.....	67
9.	Příloha	75

Seznam obrázků

Obrázek 1. Rozdělení smíšených realit podle Milgrama [2]	3
Obrázek 2. Porovnání rozšířené reality s reálným prostředím a virtuální realitou v rámci virtuálního kontinua; inspirováno [2, 12].....	9
Obrázek 3. Porovnání rozšířené virtuality s reálným prostředím a virtuální realitou v rámci virtuálního kontinua, inspirováno [2, 12].....	11
Obrázek 4. Aplikace Microsoft Translate – originální text.....	14
Obrázek 5. Aplikace Microsoft Translate – přeložený text.....	14
Obrázek 6. Ilustrace os rotace zařízení	34
Obrázek 7. Ilustrace první části detekce markeru z obrazu kamery	45
Obrázek 8. Ilustrace druhé části detekce markeru z obrazu kamery.....	45
Obrázek 9. Zpracované údaje ze senzoru pro provádění rotace	46
Obrázek 10. Žádost o povolení přístupu ke kameře v prohlížeči.....	49
Obrázek 11. Grafické uživatelské rozhraní kontroléru.....	49
Obrázek 12. Výpis z konzole serveru	50
Obrázek 13. Grafické znázornění komunikace mezi jednotlivými částmi.....	51
Obrázek 14. Ukázka úlohy Abeceda.....	52
Obrázek 15. Ukázka úlohy Pong.....	52
Obrázek 16. Ukázka úlohy Bludiště	53
Obrázek 17. Vysvětlení detekce kolizí při rotaci v úloze bludiště	54
Obrázek 18. Výpis z konzole prohlížeče při pokusu o obnovení WS spojení	56

Seznam tabulek

Tabulka 1. Porovnání vlastností video-see-through a optical-see-through přístupů	6
Tabulka 2. Porovnání rozšířené reality s virtuální realitou	11
Tabulka 3. Srovnání specializovaných zařízení pro rozšířenou realitu [38, 39, 40, 44].....	24
Tabulka 4. Podpora verzí WebGL (leden 2018) [48, 49].....	30
Tabulka 5. Podpora pro zjištění orientace a pohybu zařízení (leden 2018) [55].....	33
Tabulka 6. Shrnutí hodnot orientace zařízení	34

Tabulka 7. Podpora funkce <code>getUserMedia()</code> (leden 2018) [59]	38
Tabulka 8. Podpora WebVR (leden 2018) [62]	39
Tabulka 9. Podpora VR zařízení v prohlížečích pro hlavní OS (leden 2018) [63]	39
Tabulka 10. Podpora webových soketů (leden 2018) [67]	47
Tabulka 11. Rychlost jednotlivých částí detekce markeru v obraze, 1. část: naplnění bufferů, 2. a 3. část: jednotlivé průběhy vykreslovacím řetězcem, 4. část: přečtení dat z grafické karty, 5. část: projití přečtených dat.....	60
Tabulka 12. Shrnutí testování s různými stupni zmenšování při detekci markeru.	62
Tabulka 13. Doba potřebná pro přenos dat z kontroléru na přijímač při použití lokálního serveru	63
Tabulka 14. Doba potřebná pro přenos dat z kontroléru na přijímač při použití vzdáleného serveru.....	64

Seznam ukázek kódu

Ukázka kódu 1. Získání WebGL kontextu	30
Ukázka kódu 2. Inicializace WebGPU, převzato z [47]	32
Ukázka kódu 3. Přiřazení události pro zjištění orientace zařízení	35
Ukázka kódu 4. Přiřazení události pro zjištění pohybu zařízení.....	36
Ukázka kódu 5. Získání přístupu ke kameře v JavaScriptu.....	37
Ukázka kódu 6. Získání VR zařízení, převzato z [64]	40
Ukázka kódu 7. Návrh způsobu získání XR zařízení v rámci WebXR, převzato z [66]	41
Ukázka kódu 8. Využití rozšíření při specifikaci textury	43
Ukázka kódu 9. Implementace hlavních události pro obsluhu webových soketů....	56
Ukázka kódu 10. Implementace odesílání dat.....	57

Seznam zkratek

2D – two-dimensional space; dvourozměrný prostor

3D – three-dimensional space; trojrozměrný prostor

AJAX – Asynchronous JavaScript and XML; asynchronní způsob posílání dat v JavaScriptu pomocí XML

API – Application Programming Interface; rozhraní pro programování aplikací

AR – Augmented Reality; rozšířená realita

CD – Compact Disc; kompaktní disk

CPU – Central Processing Unit; centrální procesorová jednotka

CSS – Cascading Style Sheets; kaskádové styly, metoda stylování webových stránek

CT – Computed Tomography; výpočetní tomografie

GPS – Global Positioning System; Globální polohový systém

GPU – Graphic Processing Unit; grafická procesorová jednotka

HMD – Head-mounted display; displej nasazovaný na hlavu

HTML – HyperText Markup Language; značkovací jazyk pro hypertext

HTTP – Hypertext Transfer Protocol; protokol pro výměnu hypertextových dokumentů ve formátu HTML

ISO – International Organization for Standardization; Mezinárodní organizace pro normalizaci

JPG – formát pro ukládání obrázků se ztrátovou kompresí

JSON – JavaScript Object Notation; javascriptový objektový zápis

LCD – Liquid-crystal display; displej z tekutých krystalů

LCoS – Liquid crystal on silicon; technologie displeje z tekutých krystalů

MR – Mixed Reality; smíšená realita

MRI – Magnetic Resonance Imaging; magnetická resonance

NHL – National Hockey League; Národní hokejová liga

OCR – Optical Character Recognition; optické rozpoznávání znaků

OS – Operating System; operační systém

SDK – Software Development Kit; sada vývojových nástrojů

USD – United States dollar; Americký dolar

VR – Virtual Reality; virtuální realita

W3C – World Wide Web Consortium; konsorcium vyvíjející standardy pro web

WebGL – Web Graphics Library; webová grafická knihovna

WebVR – Web Virtual Reality; virtuální realita na webu

WS – Web Sockets; webové sokety

WWW – World Wide Web; celosvětová síť

XHR – XMLHttpRequest; asynchronní způsob přenosu dat mezi prohlížečem a serverem

XML – Extensible Markup Language; rozšiřitelný značkový jazyk

1. Úvod

Koncept rozšířené reality nepředstavuje nic nového ani převratného. První prototypy byly vytvářeny už v padesátých a šedesátých letech minulého století. Dlouho dobu se však jednalo o těžkopádné systémy, kterým se v širším měřítku nepodařilo uplatnit. Až v devadesátých letech se o rozšířenou realitu začalo zajímat více výzkumníků, protože zařízení již měla dostatek výkonu a displeje postačující rozlišení na to, aby se dokázala více prosadit. Dnes již kompatibilní zařízení vlastní většina lidí a díky tomu i aplikace mají velký potenciál se dostat mezi širokou veřejnost.

Rozšířená realita je pohled na reálné prostředí, které je rozšířeno virtuálně generovanými prvky, jakými jsou například grafika, video nebo zvuk. Je v kontrastu s virtuální realitou, která naopak kompletně nahrazuje reálný svět a zcela vtahuje uživatele do vlastního prostředí. Rozšířená realita může být považována za mezistupeň mezi reálným a virtuálním světem. Ideálně by se uživateli mělo zdát, že reálné a virtuální předměty spolu naprosto splývají a jsou nerozeznatelné.

Aplikace rozšířené reality již nyní nachází využití v mnoha situacích – například turismus, vzdělávání, nakupování a asi nejčastěji počítačové hry. S tím souvisí i různé druhy používaných zařízení. Pro základní aplikace je dostačující mobilní telefon. Zejména ve hrách a případně na specializovaných pozicích se používají specifická zařízení, která uspokojí i náročnější požadavky – kromě vyššího výkonu se často jedná o možnost mít obě ruce volné. Typickým zařízením jsou různé formy průhledných brýlí.

Práce si klade za cíl popsat možnosti rozšířené reality ve webovém prohlížeči s využitím javascriptových funkcí a WebGL. Díky takovému zacílení je možné výsledné aplikace využívat kromě klasických počítačů a notebooků také v mobilních telefonech, které z hlediska rozšířené reality představují vhodné zařízení. Nejen díky jejich masovému rozšíření ale zejména také díky splnění všech základních požadavků, které jsou pro využití rozšířené reality na zařízení kladeny

– přítomnost kamery, velký displej, dostatečný výkon a široká škála pomocných senzorů.

Text se ve své druhé kapitole věnuje základním informacím o rozšířené realitě včetně stručné historie a porovnání s virtuální realitou. Dále je popsána široká množina možných aplikací. V následující části práce prezentuje, jaká zařízení nacházejí využití v rozšířené realitě s příklady konkrétních systémů. Ve čtvrté kapitole jsou podrobně rozepsány možnosti webových prohlížečů při realizaci rozšířené reality. Pátá část se věnuje návrhu a implementaci ukázkových úloh a popisu použitých technologií a postupů. Poslední kapitola shrnuje a hodnotí výsledky, kterých bylo při implementaci dosaženo.

2. Rozšířená realita

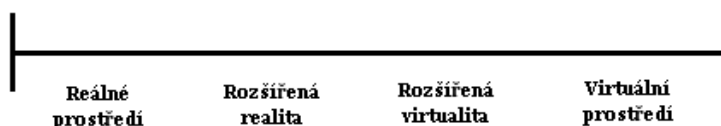
Základní princip rozšířené reality spočívá v kombinaci reálného světa a umělých objektů či jiných virtuálních informací, které se doplňují do reálného obrazu. Zásadní rozdíl mezi virtuální realitou a rozšířenou realitou je v tom, že virtuální realita vytváří kompletně virtuální obraz s účelem potlačení okolního světa, zatímco rozšířená realita se obraz reálného světa snaží rozšířit o virtuální prvky, které využívají strukturu reálného obrazu a snaží se co nejvíce do reálného světa zapadnout. Rozšířenou realitu je možné chápat i jako mezistupeň mezi běžnou realitou a virtuální realitou.

Azuma a kolektiv [1] definují rozšířenou realitu jako systém, který splňuje následující body:

- Kombinuje reálné a virtuální prvky v reálném prostředí
- Je interaktivní v reálném čase
- Registruje reálné a virtuální objekty ve vzájemném „ohledu“

Uvedená definice nelimituje využití pouze na určitý typ zařízení nebo na konkrétní lidský smysl. [1]

Na pomezí mezi reálným a virtuálním prostředím se kromě rozšířené reality nachází i rozšířená virtualita, která je založena na kompletně virtuálním prostředí, které je navíc obohaceno o informace z reálného světa. Vztah mezi jednotlivými smíšenými realitami ilustruje následující obrázek.



Obrázek 1. Rozdělení smíšených realit podle Milgrama [2]

2.1. Historie

Historie rozšířené reality sahá až do padesátých let 20. století. Morton Heilig měl vizi, že sledování filmu v kině by mohlo být něčím více než jen vizuální aktivitou. Představoval si, že by bylo možné zapojit do sledování více lidských smyslů a tím

zvýšit pocit vnoření. V roce 1955 svou myšlenku multi-senzorového kina publikoval a do roku 1962 vytvořil prototyp nazvaný „Sensorama“. Prototyp byl schopen zobrazovat stereoskopický obraz, přehrávat stereo zvuk, simulovat vítr a vypouštět do vzduchu různá aromata. [3, 4]

V roce 1968 Ivan Sutherland vytvořil během svého působení na Harvardské univerzitě zařízení nazvané „Damoklův meč“¹, které bylo předchůdcem virtuální a rozšířené reality. Jednalo se o displej nasazovaný na hlavu (HMD, „head-mounted display“), skrze který s využitím „optical-see-through“ principu (více v kapitole 2.2) bylo možné pohybem očí ovlivňovat scénu. Celé zařízení bylo značně těžké a kvůli nutnosti zaznamenávat pohyby hlavy a očí bylo zavěšeno na stropě – odtud pochází jeho název. Sám Sutherland projekt popsal projekt jako „The Ultimate Display“². [5, 6]

I v následujících desetiletí byla rozšířená realita stále spíše okrajovou záležitostí. Rozdělení realit na rozšířenou a virtuální (viz Obrázek 1) bylo definováno až v roce 1994 v článku „A taxonomy of mixed reality visual displays“³, který vydali Paul Milgram a Fumio Kishino. [2, 3]

V roce 1997 Ronald Azuma publikoval článek „A Survey of Augmented Reality“⁴, kde definoval oblast rozšířené reality, popsal problémy, se kterými je nutné počítat, a shrnul dosavadní vývoj. V článku se také pokusil nastínit možná využití rozšířené reality v medicíně, v obecné vizualizaci, v průmyslové výrobě, v robotice a ve vojenství, a problémy spojené s jednotlivými odvětvími. [1, 7]

Na konci devadesátých let začaly konat první konference zaměřené na rozšířenou realitu a vznikly první firmy, které vyvíjeli aplikace a hry, které rozšířenou realitu využívaly. ARQuake byla první venkovní mobilní hrou, kterou vytvořil v roce 2000 Bruce Thomas se svým týmem. [3, 8]

¹ v anglickém originále „The Sword of Damocles“

² česky „Ten nejlepší displej“

³ česky „Klasifikace vizuálních displejů pro smíšenou realitu“

⁴ česky „Průzkum rozšířené reality“

V následujících letech vzniká velké množství aplikací, které s rozšířenou realitou pracují, a rozšířená realita se postupně stává součástí každodenního života. Více informací o možných aplikacích včetně hry ARQuake lze nalézt v kapitole 2.7.

2.2. Principy augmentace

Z hlediska principu použitého zařízení existují 2 základní techniky augmentace – generování virtuálních prvků do reálné scény. Prvním z nich je tzv. „video-see-through“, který je blíže virtuální realitě. Generování virtuálního obrazu je ale nahrazeno obrazem reálného světa z kamery, do kterého se přidají virtuální prvky a jako celek jsou poté vykresleny na displej. Druhým principem je tzv. „optical-see-through“. Hlavním rozdílem je v tom, že reálné prostředí je vnímáno naprosto přirozeně. Dochází pouze k doplnění virtuálních prvků do reálného obrazu.

2.2.1. Video-see-through augmentace

Video-see-through technika poskytuje několik výhod. Vzhledem k faktu, že reálný obraz je digitalizován, tak je možné do něj nejen přidávat virtuální objekty, ale také z něj některé reálné objekty odstraňovat. Díky tomu je možné nahrazovat identifikační znaky (tzv. markery) virtuálními objekty. Dále technika poskytuje kontrolu nad časovou synchronizací odlišných realit a je možné lépe upravovat jas a kontrast virtuálních objektů, aby lépe zapadly v reálném prostředí. To platí zejména ve venkovním prostředí, kde se světelné podmínky mohou v jednotlivých částech dne a za různých povětrnostních podmínek velmi lišit. [9]

Mezi hlavní nevýhody patří nižší rozlišení reálného obrazu, omezený úhel zorného pole a možná dezorientace kvůli umístění kamery, které plně neodpovídá pozici očí. Problém dezorientace však může být vyřešen úpravou obrazu, aby jeho výchozí pozice odpovídala pozici očí. [9]

Typickým zařízením pro video-see-through augmentaci je neprůhledný HMD. Potřeba speciálního hardwaru je ale nepraktická a nepohodlná a pro masivnější využití je vhodné aplikace cílit takovým způsobem, aby je mohlo využívat co největší spektrum uživatelů. Proto je lepším způsobem využití mobilního telefonu, který nejen poskytuje nezbytné komponenty a vlastnosti, jakými jsou displej

s dostatečným rozlišením, kamera a možnost intuitivního ovládání, ale také ho dnes vlastní téměř každý.

2.2.2. Optical-see-through augmentace

Optical-see-through princip typicky využívá průhledné brýle, díky čemuž zůstává rozlišení reálného obrazu netknuté, protože je očima vnímáno zcela přirozeně. Využití nachází zejména v situacích, kdy je nezbytné, aby stále bylo možné vnímat reálný svět i při poruše zařízení – například při využití v medicíně během operativních zákroků, kde by případná porucha mohla i ohrožovat pacienty na životě. [9]

I když se mohlo zdát, že jde o lepší a přirozenější přístup, tak při vykreslování virtuální části informace je nutné navíc dbát i na synchronizaci s reálnou částí, nad kterou nemá z principu daného přístupu vykreslovací řetězec žádný vliv.

Navíc kombinace virtuálních objektů s reálným obrazem skrz průhledné sklo nebo čočky snižuje jas a kontrast jak virtuálních, tak i reálných objektů, kvůli čemuž je technika méně vhodná pro venkovní využití. Z principu použití brýlí je zorný úhel omezený a objekty doplněné v rozích se mohou jevit deformované. [9]

Tabulka 1. Porovnání vlastností video-see-through a optical-see-through přístupů

	Video-see-through	Optical -see-through
Rozlišení reálného obrazu	Snížené	Přirozené
Jas a kontrast virtuálních prvků	Lepší	Horší
Zorný úhel	Omezený	Omezený
Prostředí	Bez omezení	Možné omezení ve venkovním prostředí
Možnost odstranění reálných objektů	Ano	Ne
Dezorientace uživatele	Může nastat Lze řešit úpravou obrazu	Není
Stav při poruše zařízení	Kompletní ztráta vidění	Ztráta pouze virtuálních informací

2.3. Druhy augmentace

Augmentace okolí se nemusí omezovat pouze na obraz ve smyslu obrazců a objektů. Doplněvaná informace může mít podobu textu vykresleného na označeném povrchu. Virtuální textová informace by mohla těžit z faktu, že její zobrazování je plně pod kontrolou aplikace, která díky tomu může uživatelům nabízet lokalizované texty.

Informace ale nemusí být jen vizuálního charakteru. Není obtížné si představit situaci, kdy by bylo vhodné doplnit zvukovou informaci ať už podle aktuální polohy nebo směru pohledu. V případě interakce s virtuálními objekty je možné doplňovat zvuk, který by takovou interakci uživateli ještě lépe charakterizoval a učinil ji více reálnou.

Při použití dodatečných mechanismů si lze s trochou fantazie představit i využití pachové augmentace. Forma hmatové augmentace by mohla pro nevidomé uživatele fungovat jako jakási forma vizuální augmentace pro běžné uživatele.

2.4. Mobilní rozšířená realita

Mobilní rozšířená realita se vyznačuje aplikací rozšířené reality v mobilním prostředí mimo předem připravené prostory. Dále je charakteristická využitím bezdrátové komunikace, principů lokalizace a služeb, které lokalizaci zajišťují (např. GPS). Z toho vycházejí i vyšší požadavky na zařízení. Klasické zařízení s kamerou musí být rozšířeno o schopnost komunikace, příjem lokalizačních signálů a pokud možno i doplňkové senzory, které zajistí informaci o poloze zařízení. Jako charakteristické zařízení si lze představit mobilní telefon. Jeho typické využití vypadá tak, že uživatel namíří kamerou na místo zájmu, její vstup je doplněn virtuálními objekty a posléze jsou jako celek zobrazeny na displeji. [10]

Hlavní rozdíl od obyčejné rozšířené reality spočívá v tom, že mobilní rozšířená realita je rozšířena o dimenze času a prostoru. To znamená, že je více zaměřena na celé prostředí než na jeden či jednotky fixních objektů a že je orientována na využití v rámci delšího časového úseku. Kvůli tomu aplikace mobilní rozšířené reality čelí problémům se získáváním informací v reálném čase, s dynamickým

rozeznáváním objektů a se sledováním pohybu. Díky tomu je ale uživatelům dopřána mnohem vyšší úroveň svobodného pohybu. [10]

Mobilní rozšířená realita má velký potenciál a díky stále rostoucímu rozšíření mobilních telefonů, které poskytují dostatek výkonu v relativně malé velikosti zařízení, se stává stále použitelnější v každodenním životě [12]. Díky její flexibilitě je možné vytvářet aplikace, které plně využívají okolní prostředí uživatele [13].

S využíváním okolního prostředí je ale spojeno několik společenských problémů, které bude nutné před plnohodnotným využitím mobilní rozšířené reality v běžném životě vyřešit. Jedná se o problémy s minimalizací obtěžování okolí uživatele, s přirozenou interakcí se systémem a s ochranou soukromých informací. Mobilní telefony představují rušivý element nejen pro okolí vlastníka, ale často i pro jeho samotného. Jedná se například o hlasité telefonování v uzavřených prostorech, jakými jsou autobusy nebo společné pracovní místnosti. Stejnému problému bude čelit i využívání aplikací rozšířené reality. Při jejich návrhu je tedy nutné dbát na to, aby interakce byla co nejvíce diskrétní a nenápadná. Systém, který nebude rušit osoby v okolí, se pak může snadno stát společensky akceptovatelným. Správně navržený systém nebude rušit ani svého uživatele, pokud mu nebude do přímého zorného pole vnucovat méně důležité informace. [3]

V případě pohybu na veřejných prostranstvích je dalším důležitým faktorem způsob interakce se zařízením. Interakce by měla být co nejvíce přirozená, protože pokud by byla nějakým způsobem nepřirozená, tak by na veřejnosti vzbuzovala příliš mnoho pozornosti, čímž by se stávala příliš rušivou. Jedním z možných řešení je použití náramku, který v sobě obsahuje nutná zařízení k rozpoznání pohybu a polohy, a provádět ovládání gesty. Uživatel by s objekty a informacemi ve scéně interagoval pomocí pohybů ruky. Dřívější způsoby ovládání pomocí hlasu nedosahovaly takové přesnosti, byly nepoužitelné na rušných místech a navíc byly i společensky těžko přijatelné. [3]

Systémy mobilní rozšířené reality musí také vyřešit problém práce se soukromými informacemi na veřejných místech. To znamená, že informace by měl vidět pouze uživatel a ostatní by je měli vidět jen v situaci, kdy to uživatel dovolí. Existují návrhy speciálních brýlí a kontaktních čoček, které by soukromí dokázaly

poskytnout. Zároveň by taková forma zařízení byla velmi nenápadná a tudíž společensky snadno akceptovatelná. [3]

2.5. Porovnání s virtuální realitou

Virtuální realita je druhem smíšené reality, která je naprostým protipólem vůči reálnému prostředí [12]. [11] definuje virtuální realitu jako umělé prostředí, které je vnímáno smysly, které jsou stimulovány počítačem, a ve kterém uživateli akce částečně ovlivňují, co se v prostředí stane. [9] dále definuje virtuální realitu jako počítačem vygenerované prostředí, se kterým je možné interagovat jako by bylo reálné. Následující obrázek (Obrázek 2) zobrazuje grafické srovnání rozšířené reality s reálným prostředím a virtuální realitou. Z reálného prostředí je vyjmuta konvička a je nahrazena virtuálním protějškem.



Obrázek 2. Porovnání rozšířené reality s reálným prostředím a virtuální realitou v rámci virtuálního kontinua; inspirováno [2, 12]

Cílem virtuální reality je kompletní nahrazení reálných vjemů uživatele. Zpravidla se jedná o vizuální a zvukovou simulaci. Na druhé straně stojí rozšířená realita, která se také snaží o simulaci vizuálních a zvukových vjemů. Liší se ale v tom, že nedochází ke kompletnímu nahrazení reálného světa, ale k jeho doplnění [14, 16]. Rozšířená realita může také odstraňovat části reálného světa, ať už se jedná o odstraňování objektů nebo potlačování některých zvuků, a provádět jejich nahrazení virtuálními informacemi.

Virtuální realita typicky využívá speciální neprůhledné brýle s displejem uvnitř, které si uživatel nasadí, aby byla potlačena okolní realita. Naopak rozšířená realita ke svému fungování obecně vyžaduje jen zařízení s displejem a navíc s kamerou, protože program, který doplňuje virtuální informace, je musí správně zasazovat

v kontextu reálných objektů [14]. Na rozdíl od virtuální reality nevyžaduje, aby zařízení zajišťovalo i odtržení od reálného světa, což ji činí méně náročnou na zprovoznění [15]. Dnešní typické zařízení je mobilní telefon, protože v sobě zahrnuje jak displej, tak kameru [14]. I rozšířená realita může využívat brýle, které mohou být jak průhledné tak neprůhledné. Nošení průhledných brýlí s kamerou je oproti mobilnímu telefonu v současné době pravděpodobně nejpohodlnější způsob využití rozšířené reality, protože uživatel stále zůstává volné ruce, které nemusí mířit telefonem na místo zájmu. V takovém případě by se náklady na pořízení blížily zařízením vyžadovaných virtuální realitou.

Virtuální realita poskytuje naprosté ponoření do svého obsahu, proto je v zájmu daného zařízení, aby se uživatel co nejvíce odpoutal od reálného světa. V kontrastu s tím je rozšířená realita, která naopak upoutává pozornost na reálný svět s virtuálními prvky, které by ale ideálně měly být nerozeznatelné. [15]

Rozšířená realita klade na zařízení i vyšší výpočetní nároky, protože kromě samotného vykreslování musí provádět analýzu obrazu, aby doplněné prvky naprosto hladce zapadly. Navíc by virtuální prvky měly být vytvořeny v takové kvalitě, aby nebylo možné poznat, že v původním obraze nebyly. V tom je hlavní rozdíl proti virtuální realitě, kde je veškerý vykreslovaný obsah pod kontrolou aplikace a nezáleží tolik na tom, že výsledný obraz není fotorealistický. Doplněné objekty v rámci rozšířené reality jsou nutně porovnávány s reálným okolním prostředím, což klade mnohem vyšší nároky na kvalitu renderování.

Typická využití jednotlivých technologií vycházejí zejména z úrovně vnoření, kterou poskytují. Aplikace virtuální reality se hodí v situacích, kdy je nutné uživatele kompletně zaujmout virtuálním světem – například značné množství her. Naopak rozšířená realita nachází uplatnění typicky v situacích, kdy je vhodné uživatele neodtrhávat od reálného světa a informace do něj pouze doplnit.

Následující tabulka poskytuje shrnutí srovnání rozšířené reality s virtuální realitou. Tabulka se omezuje na heslovité shrnutí informací, které jsou popsány v textu výše.

Tabulka 2. Porovnání rozšířené reality s virtuální realitou

	Virtuální realita	Rozšířená realita
Stav vůči reálnému světu	Úplně nahrazení	Doplnění nebo částečné nahrazení
Náročnost na pořízení	Vyšší	V základu nižší
Nezbytný hardware	Displej	Displej a kamera
Úroveň pohlcení	Naprosté	Částečné
Výpočetní náročnost	-	Častěji vyšší

2.6. Rozšířená virtualita

Rozšířená virtualita je protějškem k rozšířené realitě. Stojí na pomezí mezi virtuální a rozšířenou realitou a stejně jako v rozšířené realitě dochází ke kombinaci virtuálních a reálných prvků. Rozdíl je v tom, v jakém prostředí probíhá interakce. Probíhá-li ve virtuálním světě, tak se jedná o rozšířenou virtualitu.

Je možné ji také popsat jako podmnožinu smíšené reality, která má blíže k virtuální realitě než k reálnému prostředí. Principiálně vychází z virtuální reality, která je obohacena o informace z reálného světa. Rozšíření nemusí být realizováno pouze formou reálných objektů ve virtuální scéně. Je možné také využít příjem informací ze senzorů, který mohou například být využity objekty ve virtuální scéně k provádění transformací.

Následující obrázek (Obrázek 3) zobrazuje grafické srovnání rozšířené virtuality s reálným prostředím a virtuální realitou. Z reálného prostředí je převzata konvička a je zasazena do jinak kompletně virtuálního prostředí.



Obrázek 3. Porovnání rozšířené virtuality s reálným prostředím a virtuální realitou v rámci virtuálního kontinua, inspirováno [2, 12]

Příkladem aplikace využívající principů rozšířené virtuality je „Windows on the World“, která byla vytvořena v roce 1999. Cílem práce bylo vytvořit virtuální svět, který je rozšířen obrazem z reálného světa – například odrazy v oknech, počítačové obrazovky nebo tabule. Implementovaný systém je schopen využívat vstupů z kamery a její obraz mapovat jako texturu ve vygenerované virtuální scéně. [17]

Na podobném principu implementoval rozšířenou virtualitu v roce 2001 i tým v Japonsku [18].

2.7. Aplikace rozšířené reality

Rozšířená realita již nyní nachází využití v celé řadě odvětví. Konkrétní aplikace se v různých situacích různí, ať už se jedná o využitý princip augmentace nebo o požadavky konkrétního využití. V současné době se rozšířená realita uchycuje a výzkum v některých oblastech stále ještě intenzivně probíhá. Je tedy pravděpodobné, že v budoucnu se rozšířená realita prosadí i do dalších oblastí, ať už se bude jednat o usnadnění všedního života nebo zvýšení efektivity řešení problémů ve sféře různých specializovaných profesí.

2.7.1. Zdravotnictví

Jedním z možných využití ve zdravotnictví je příprava údajů o pacientovi z vyšetření, jakými jsou například magnetická resonance nebo výpočetní tomografie. Získané údaje mohou posloužit pro následnou projekci na tělo pacienta při provádění drobných invazivních zákroků, při kterých je kladen důraz na to, aby byl vnější řez co nejnepatrnější. Problém při těchto vyšetřeních spočívá v tom, že lékařova schopnost vidět dovnitř pacienta je omezená. Využití například speciálních brýlí s rozšířenou realitou, které by poskytovaly vhled pod pacientovu kůži, by dokázalo invazivnost daného zákroku ještě více snížit. [7]

Dalším využitím by mohla být výuka chirurgů nebo zefektivnění tréninku nových postupů. Virtuální instrukce by mohly sloužit jako nápověda správného postupu při provádění zákroku. Případně by obraz mohl být přenášen za účelem obdržení odborné vzdálené asistence. K zamyšlení by jistě bylo, zda by takové informace nebyly příliš narušující pro průběh zákroku. Možným řešením by mohlo být

zobrazování instrukcí pouze na vyžádání. Vzhledem k povaze povolání chirurga by bezchybné hlasové ovládání bylo nezbytností.

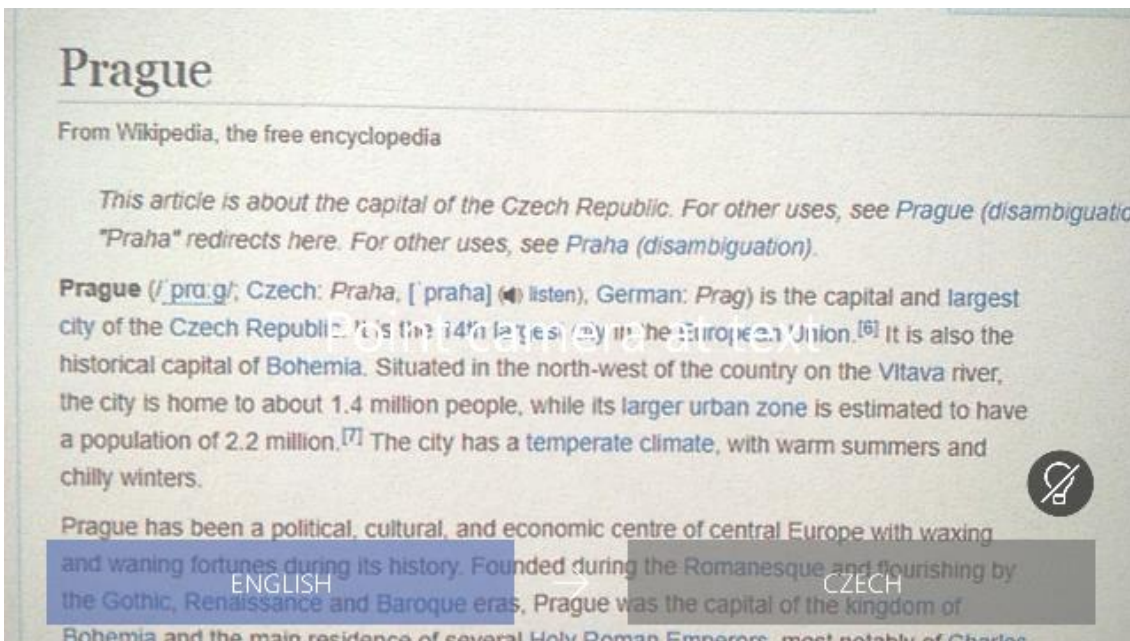
2.7.2. Turismus

Při cestování je jednou z hlavních překážek neznalost místního jazyka. Ačkoliv je možné texty ručně přepisovat do překladače, tak se jedná pomalý způsob získání překladu. Navíc v případě některých jazyků, jakými jsou například čínština nebo japonština, je pro většinu lidí učinit přepis téměř nemožné. Je to situace, kterou rozšířená realita dokáže společně s OCR technologií velmi usnadnit.

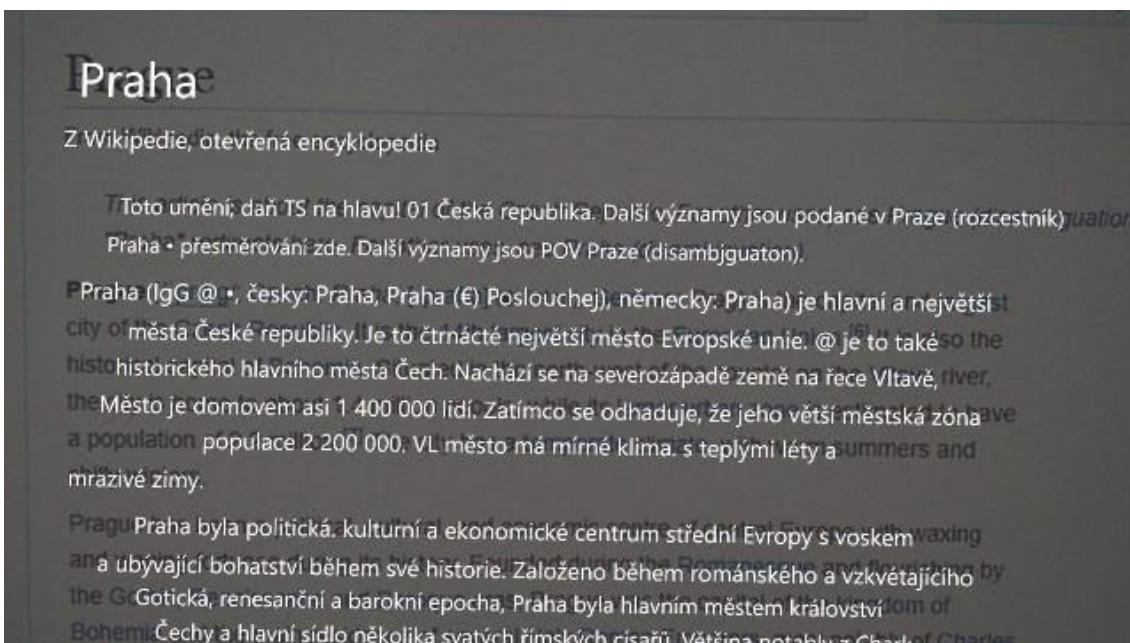
Jednou z prvních aplikací tohoto typu bylo „Word Lens“. Fungovalo na principu skenování textu pomocí kamery v mobilním telefonu a následné identifikaci textu a jeho překladu [19]. Přeložený text se zobrazoval v původním kontextu na původním pozadí. Zvoleným přístupem se aplikace snažila maximálně využít aktuální situaci na trhu, kde kompatibilní mobilní telefon vlastnilo značné množství lidí. V roce 2014 společnost, která aplikaci vyvinula, koupil Google a o rok později získanou technologii zahrnul do svého překladače [19].

V současné době je pro platformy Android a iOS dostupná aplikace „Google Translate“ (česky Google překladač) a pro Android, iOS a Windows je k dispozici aplikace „Microsoft Translate“ (česky Microsoft překladač). Obě aplikace fungují na podobném principu. Základní využití vypadá tak, že se kamerou zařízení namíří na text v cizím jazyce a aplikace místo původního textu vloží do obrazu překlad. Kromě překladu textu v obrazu obě aplikace umožňují i překlad hlasu a překlad ručně vloženého textu. [20, 21]

Následující dva obrázky (Obrázek 4, Obrázek 5) ilustrují překlad pomocí aplikace Microsoft Translate. První z nich obsahuje originální text, na který je hleděno skrze aplikaci a obsahuje základní pokyny pro ovládání. Druhý obrázek ukazuje výslednou podobu přeloženého textu, který překrývá text původní. Ukázkovým textem je začátek článku o Praze na anglické verzi Wikipedie.



Obrázek 4. Aplikace Microsoft Translate – originální text



Obrázek 5. Aplikace Microsoft Translate – přeložený text

Na několika místech na světě je turistům nabízen systém, který jim zobrazuje informace o navštívených místech a budovách, když se na ně přes zapůjčené brýle podívají. Aplikace se neomezuje pouze na prosté sdělování doplňujících informací, které se mohou dozvědět i od průvodce, ale také umožňuje prohlédnout si budovy a krajinu v původní podobě, jak vypadaly v minulosti [22]. Příkladem takové

aplikace je projekt „Archeoguide“. Jeho cílem je poskytnout historickým památkám mobilního průvodce, který by návštěvníkům dokázal sdělovat informace o archeologických nalezištích. Motivací pro jeho vývoj bylo nahrazení klasických audio průvodců, které neposkytují přesnou navigaci a schází jim uživatelská přívětivost [12, 23].

2.7.3. Vzdělávání

Rozšířená realita může při výuce ve školách fungovat jako doplněk k materiálům z učebnic. Speciální stránky v učebnicích historie mohou sloužit jako spouštěcí znak pro přehrání historické události nebo například k prohlédnutí trojrozměrného modelu budovy či města v daném období [22].

Vlády některých států se rozhodují k investicím do vzdělávání, které mají podpořit využívání rozšířené a virtuální reality při výuce. Patří mezi ně například Francie. Cílem technologických předmětů na středních školách má být poskytnutí studentům zkušenosti s řešením problémů pomocí virtuální a rozšířené reality. Studenti jsou podněcováni, aby pomocí technologií řešili nastalé problémy a navrhovali jejich řešení. [24]

V Jižní Koreji se vláda v roce 2016 rozhodla v následujících 5 letech investovat 363 milionu amerických dolarů (v přepočtu přibližně 8 miliard českých korun) na vybavení a podporu výzkumu v rámci rozšířené a virtuální reality [24].

2.7.4. Nakupování

Online nakupování je dnes běžnou praxí. V případě objednávání oblečení z internetu čelí uživatelé problému, že si vybraný kus nemohou vyzkoušet. „Magic Mirror“ je technologie, která se snaží uživatelům umožnit si virtuálně vyzkoušet boty, které by si rádi koupili. Základem principu jsou speciální ponožky, které se svými připravenými značkami slouží sledovacímu systému, aby správně do obrazu vložil „zkoušené“ boty. Uživatel stojí před LCD obrazovkou, která funguje jako zrcadlo a vidí v ní sám sebe s virtuálně doplněnými botami. [3]

Podobný problém jako nákup oblečení obnáší i zařizování interiérů. Pro mnoho lidí může být obtížné si představit, jak by vybraná skříň či pohovka vypadaly

v kontextu místnosti, kterou se snaží vybavit. S pomocí mobilní aplikace, která by nabízela možnost si různé kusy nábytku prohlédnout a virtuálně umístit na reálné požadované místo, by jistě zařízení místnosti probíhalo pohodlněji, protože do obchodu by člověk už šel s jasnou představou toho, co chce a jakou barvu by vybraný kus nábytku měl mít. Zároveň by však mohl být úplně eliminován prvek osobní návštěvy kamenného obchodu a v aplikaci by mohlo probíhat i samotné objednání.

Celý princip usnadnění nakupování s pomocí rozšířené reality spočívá v tom, že zákazníci si nemusí nic zkoušet přímo v obchodech a mohou tak ušetřit značné množství času, který by byl potřeba na dopravení se do obchodu [3]. Další výhodou může být fakt, že v obchodě nemusí být naskladněno veškeré zboží ve všech velikostech a možnost zjištění dané informace již doma taktéž sníží množství ztraceného času. Naopak zákazník může být buď odkázán na prodejnu, kde je jím vybraný kus k dispozici, nebo provést objednávku z domova přímo z centrálního skladu.

Aplikace typu „Magic Mirror“ stále ještě nedosáhly plného nasazení, protože stále existují problémy, které je nutné vyřešit. Aby mohly obstát před širokou veřejností, tak musí poskytovat bezchybnou detekci značek, aby byl uživatel přesvědčen, že virtuálně doplněné oblečení vypadá přesně tak, jak bude nakonec vypadat, až si ho skutečně obleče. [3]

2.7.5. Televizní přenosy a reklamy

Rozšířená realita nachází využití i při vysílání sportovních přenosů. Přenášený obraz je před odesláním v reálném čase obohacován o informace, které se snaží divákovi usnadnit sledování. V případě závodních přenosů mohou ukazatele v obraze zobrazovat zřetelnou identifikaci vozidla a další informace například o dosažených časech. Stejně tak například fotbalové přenosy mohou zobrazovat hranice ofsajdů a dalších herních situací. [1]

Jedním z prvních vyvinutých systémů je „FoxTrax“. Jedná se o soustavu kamer a vysílačů sloužící ke sledování a zvýrazňování hokejového puku, který je často

obtížné sledovat na televizních obrazovkách. Motivací pro jeho vývoj bylo získání více diváků ke sledování přenosů NHL. [25]

Úpravy přenášeného obrazu se nemusí týkat jen zlepšování diváckého zážitku. Na stejném principu může fungovat i vyplňování reklamy na připravených plochách [1]. Je snadné si představit situace, kdy různá reklama je různým divákům nebo případně stejná reklama v různých jazykových verzích vysílána do různých částí světa při mezinárodních přenosech.

Jedna z dalších možných podob reklamy je vytištění identifikačního znaku v časopisu. Například automobilka MINI provedla v roce 2008 marketingovou kampaň v několika německých automobilových časopisech. Čtenář časopisu mohl speciální stránku zobrazit na webkameru na webové stránce automobilky a 3D model nového modelu MINI se s využitím technik rozšířené reality zobrazil na stránce časopisu podle jejího natočení [3, 26]. Zmiňovaná webová stránka již dnes není k dispozici.

2.7.6. Počítačové hry

Zábavní průmysl je oblast, ve které rozšířená i virtuální realita již dnes nacházejí široké využití. Jedním z důvodů je například zvýšená úroveň pohlcení v případě virtuální reality a nadšení z možnosti využití reálného okolí v případě rozšířené reality. Následující výběr je soustředěn na hry, kterým se nějakým způsobem daří vyčnívat nad ostatní a dostaly se tak do širšího povědomí ať už svou originalitou nebo jen prostým znovupoužitím již známé značky.

2.7.6.1. ARQuake

ARQuake bylo první hrou, která využila principů rozšířené reality ve venkovním prostředí s využitím mobilního systému. Jedná se o rozšířenou verzi hry Quake upravenou pro použití s rozšířenou realitou. Quake je typem tzv. „střílečka z pohledu první osoby“ (z anglického „first-person shooter“). Hlavním rozdílem je odstranění celého prostředí původní hry, protože je nahrazeno reálným prostředím kampusu školy, kde byla hra vyvinuta. Monstra zůstávají původní pouze se změněnými barvami, aby více odpovídaly jinému prostředí. Hráč se musí aktivně v prostředí pohybovat. Jeho pozice je získávána pomocí GPS,

elektronického kompasu a inklinometru, aby hra přesně věděla, kde se nachází. GPS senzor využívá speciální přijímač, který dosahuje přesnosti až 50 centimetrů s aktualizacemi 10 krát za sekundu, na rozdíl od běžného, který poskytuje pouze přesnost do 5 metrů a 1 aktualizaci údajů za sekundu. [8, 27]

Autoři popisují několik problémů, které museli během vytváření hry vyřešit. Při zobrazování virtuálních objektů v reálném prostředí je nutné věnovat zvláštní pozornost barvám virtuálních objektů, aby byly dobře viditelné za různých světelných podmínek, které se ve venkovním prostředí mohou na různých místech a v různých částech dne velmi lišit. Dále bylo nutné modifikovat původní hru tak, aby klávesové vstupy indikující pohyb v prostředí byly nahrazeny vlastním pohybem hráče, s čímž souvisí i nutná kalibrace herního souřadnicového systému. Dále musel být vyvinut systém poskytující 6 stupňů volnosti s dostatečnou přesností při nízké ceně. Celý systém navíc musí být dostatečně kompaktní, aby se s ním hráč mohl volně pohybovat. [8]

Z dnešního pohledu byl systém příliš velký, což muselo značně zneprůjemňovat jeho nošení. Aby se uživatel mohl pohybovat, musel na sobě nosit veškeré přístroje potřebné ke hraní včetně baterií nutných k jejich napájení. Vzhledem k tomu, že hra byla vyvinuta v roce 2000, tak nebylo možné využít miniaturizace jednotlivých prvků, kterou dnes využívají moderní mobilní telefony. Při odstranění speciálních brýlí a zbraně se ale jedná o podobný soubor součástek.

2.7.6.2. Pokémon Go

Pravděpodobně nejznámější příklad hry s prvky rozšířené reality je v současné době „Pokémon Go“, která jen měsíc po svém vydání překonala 5 světových rekordů – například nejstahovanější hra s dosaženými 130 miliony staženími během prvního měsíce nebo mobilní hra s nejvyššími výnosy měsíc po svém vydání (206,5 milionu amerických dolarů, přibližně 4,6 miliardy českých korun) [28].

Hra je ve formě mobilní aplikace, čímž je dostupná téměř každému, kdo vlastní podporovaný mobilní telefon. Hlavním principem hry je pohyb po reálném světě, kde jsou podle GPS souřadnic umístěny herní předměty. Hra nabízí 2 režimy –

první slouží pouze k virtuálnímu nadhledu s umístěnými prvky zájmu a druhý využívá kameru zařízení, skrze kterou se lze dívat na reálný svět, kam jsou doplněny virtuální objekty. Tyto však s reálným prostředím nijak neinteragují.

Aplikace vzbuzovala značnou kontroverzi, protože jako místa zájmu byly zvoleny i památníky, muzea či hřbitovy. Mezi konkrétní místa patřily například Muzeum holocaustu v Osvětimi v Polsku a Památník míru v Hirošimě v Japonsku. Provozovatel hry některá místa následně ze hry odstranil [30, 31]. Na druhou stranu mnoho komerčních subjektů, jakými jsou například restaurace, se snažily aktivně propagovat, že z jejich prostor jsou herní místa zájmu dostupná [29].

Problémem nejen hry Pokémon Go ale obecně her pracujících s rozšířenou realitou je ten, že na veřejných prostranstvích vtahují do děje nejen samotné hráče, ale vzhledem k tomu, že pracují s reálným okolním prostředím, tak nutně ovlivňují i okolní osoby. Dnes neexistují žádná pravidla upřesňující, které prostory mohou být využity k augmentaci v rámci cizích aplikací. Pokémon Go nepředstavuje hru, která by zahrnovala nový dosud neviděný herní mechanismus. Ale její úspěch názorně ukázal problematické situace, se kterými se budoucí aplikace budou muset vypořádat. [29]

2.7.6.3. Ingress

Ingress je mobilní hra využívající principů rozšířené reality. Podobně jako Pokémon Go je založena na lokalizaci a pohybu v reálném světě. Hra byla poprvé zveřejněna v listopadu 2013 pro mobilní systém Android. Verze pro iOS byla vydána v červenci 2014. [32]

Hráči jsou ve hře rozděleni do dvou skupin a jejich základním cílem je obsazování portálů, které se po splnění daných kritérií spojují do trojúhelníkových polí. Podle počtu vytvořených polí se pak odvíjí, kolik bodů skupiny dostávají. Zajímavé je, že nezáleží na velikosti pole, ale na to tom, kolik lidí se v poli nachází. Spojnice mezi portály mohou být libovolně dlouhé, od jednotek metrů až po stovky kilometrů. Pozice portálů jsou vybrány tak, aby se nacházeli na reálných pozicích různých staveb – památníky, knihovny, dopravní terminály, parky apod. – a jejich množství ve hře se neustále navyšuje. Pozice portálů spolu s informací, které skupině

momentálně patří, může hráč sledovat na mapě. Aby mohl s portálem interagovat, tak musí být fyzicky v jeho těsné blízkosti. [32]

2.7.6.4. SpecTrek

SpecTrek je hra navržená tak, aby podporovala aktivní fyzické cvičení. Sloganem hry je „zachraň svět, zůstaň ve formě“ (z anglického „protect the world, stay in shape“). Její hlavní náplní je lovení duchů na základě jejich reálné polohy. Hráč na mapě vidí všechny duchy, kteří se nachází v daném poloměru. Obtížnost hry je určena časem a lze si vybrat tři různé úrovně – 15 minut, 30 minut nebo 60 minut. Hráč vidí duchy skrze kameru mobilního telefonu a v dostatečné blízkosti je může chytit. [34]

Aplikace se v roce 2009 umístila na druhém místě v soutěži „Android Developers Challenge II“ v kategorii „Životní styl“. [33]

2.7.6.5. LyteShot

LyteShot je mobilní herní systém, který se skládá z doplňkových periférií, které jsou připojeny k mobilnímu telefonu, který obstarává komunikaci se vzdáleným serverem. V základní sestavě se nachází tzv. „Lyter“, což je ruční zařízení, které slouží k zaměřování a vysílání signálu, a „LytePuck“, který je přepevný na těle hráče a slouží k příjmu signálu. Zařízení mezi sebou komunikují pomocí infračerveného záření a s mobilním telefonem pomocí Bluetooth. Základní herní situace vypadá tak, že jeden z hráčů „zasáhne“ svým Lyterem protivníkův LytePuck, který začne vydávat zvuk, aby oznámil zásah. Zároveň je informace o zásahu předána mobilnímu telefonu, který ji dále přepoše na centrální server. Tam se informace uloží a je dále odeslána prvnímu hráči, který se tak dozví o úspěšném zásahu. [35]

Mark Ladd a Tom Ketola, spoluzakladatelé společnosti LyteShot, v rozhovoru pro web Gizorama uvedli, že princip hry se značně liší o ostatní her založených na lokalizaci. Většina her sdílí podobný princip jako například Ingress. Jsou to hry, které vyžadují, aby se v oblasti, kde hra probíhá, pohybovalo velké množství hráčů. Jedná se o globální hry. Naopak LyteShot se hraje lokálně a jednotlivé hry jsou časově relativně krátké. [36]

Celá platforma je otevřená, což umožňuje vytvářet další rozšíření. S využitím 3D tiskárny lze navrhnout a vytisknout vlastní periferie a doplňkové komponenty, se kterými je pak možné hrát. Základní komponenty využívají Arduino, takže jakékoliv další zařízení s podporou Bluetooth s nimi může komunikovat. Jejich specifikace jsou volně k dispozici, díky čemuž je možné si je také vytisknout. [37]

Vývojáři mohou využívat dostupnou sadu vývojových nástrojů (SDK) k návrhu vlastních her s využitím herního engine Unity 3D. Systém umí také využívat brýle s rozšířenou realitou. [35]

2.7.6.6. Adaptivní hra s virtuálními protihráči

Počítačové hry zasazené v reálném prostředí mají předpoklady i pro společné hraní reálných a virtuálních hráčů. Nabízí se například hra na způsob „laser tagu“, kdy by lidští hráči hráli proti virtuálním, kteří by byli všem simultánně promítáni z centrálního serveru skrze speciální brýle, které by hráči museli mít. Díky takovému využití by bylo snadné upravovat složitost hry podle dovednosti skupiny nebo podle jejich věku. Centrální server by mohl jednat i adaptivně. Taková hra mohla vznést i etickou otázku, do jaké míry je vhodné, aby virtuální hráč ovládaný centrálním počítačem měl za cíl zneškodnit své lidské protivníky, obzvláště pak v případě pokud by virtuální protivníci byli řízeni nějakou formou umělé inteligence.

2.7.7. Budoucí aplikace

Vzhledem k množství různých aplikací lze usoudit, že využití rozšířené reality je velmi rozmanité a dokáže ovlivnit mnoho životních situací. Stále ale existuje mnoho oblastí, kde by rozšířená realita mohla najít výborné využití, které ale v současné době není možné aplikace implementovat kvůli omezením hardwaru a algoritmů. [22]

3. Hardware pro rozšířenou realitu

Nezbytný hardware pro implementaci rozšířené reality jsou displej pro zobrazování a kamera pro snímání okolního světa. Výsledné zařízení může mít různou podobu. Nejjednodušeji představitelným zařízením je mobilní telefon. Pokročilejší zařízení jsou ve formě průhledných brýlí, které jsou zároveň i displejem, do kterého jsou vykreslovány virtuální prvky. Oproti mobilnímu telefonu je zde výhoda v tom, že uživatel se nemusí zabývat ručním ovládáním, celkově je mnohem přirozenější se pouze dívat na místo zájmu.

Mobilní telefony představují v dnešní době ideální zařízení z pohledu rozšířené reality. Kromě toho, že zabudovaná kamera je naprostým standardem, tak také mají dostatek výkonu a navíc jsou lehké a všudypřítomné. Většina modelů má i dostatečně velký a kvalitní dotykový displej, který je nutný nejen pro dostatečně ostré zobrazení, ale také pro samotné ovládání. Dále v sobě obsahují i širokou škálu senzorů, které poskytují další informace o okolním světě a stavu zařízení. Nejdůležitější je ale právě jejich všudypřítomnost, díky které je možné již dnes vyvíjet aplikace využívající rozšířenou realitu, které mají potenciál se dostat mezi široké množství lidí.

Síťová konektivita představuje nedílnou součást dnešního života a je důležitá pro mnoho aplikací. Pro aplikaci, která je spuštěna ve webovém prohlížeči, je z principu připojení k internetu nezbytné. Všudypřítomnost internetu se navíc stále prohlubuje. Téměř většina dnešních moderních zařízení se umí připojovat k Wi-Fi sítím a mobilní zařízení navíc mohou využívat sítí mobilních operátorů pro přístup k internetu. Připojení pak je možné sdílet přes Wi-Fi, aby se mohla připojit i zařízení, která nejsou schopna připojení do mobilní sítě – zejména notebooky. Díky všem těmto faktům je možné bez obav vyvíjet aplikace ve formě pro webové prohlížeče.

Díky zacílení na webový prohlížeč bude program fungovat i na tabletech, notebookech a desktopech, pokud zařízení splní základní požadavek – má buď vestavěnou, nebo jiným způsobem připojenou kameru.

3.1. Senzory

Sensor je elektronická součástka, která umí detekovat události nebo změny v okolním prostředí a tyto informace převádí na signál, se kterým je možné dále pracovat. Při implementaci rozšířené reality se jedná o naprosto nezbytné údaje, bez kterých by nebylo možné zjistit dostatek informací o okolí a poskytnout tak přesvědčivý uživatelský zážitek. Mobilní telefony už v základu v sobě často mají širokou škálu různých senzorů.

Akcelerometr slouží k měření akcelerace, která na zařízení působí relativně vůči volnému pádu. Zároveň slouží i k detekci orientace zařízení, kde typickým využitím je otáčení obrazovky podle natočení daného zařízení. Gyroskop je sensor, který také dokáže určit orientaci zařízení – oproti akcelerometru však s vyšší přesností.

Magnetometr dokáže detekovat magnetická pole. Pokud není v blízkosti zařízení lokální zdroj magnetického pole, tak dokáže určit směr k magnetickému pólu Země.

Sensor okolního osvětlení měří jasnost okolního světla. Zpravidla se využívá k automatické adaptaci jasu displeje zařízení. Za jasného dne při pohybu venku je nutné, aby displej zářil co nejvíce a byl tak pořád pohodlně čitelný, a zároveň je zbytečné, aby v naprosto tmavém okolí displej naplno zářil, což by nejen zbytečně vybíjelo baterii, ale také nadměrně namáhalo zrak uživatele.

Sensor přiblížení slouží v mobilním telefonu typicky k vypnutí displeje při telefonování, aby nedocházelo k nechtěnému ovládní přiloženého telefonu k uchu. V dalších zařízeních může být využito k varování uživatele, že se blíží nějaká překážka. V případě rozšířené reality se o nutnost nejedná, ale uživatel plně ponořen do obsahu by také mohl přehlédnout zjevnou překážku. Nezbytné využití nachází zejména v zařízeních pro virtuální realitu.

Senzorů samozřejmě existuje mnohem více, ale jejich využitelnost stejně jako jejich rozšíření je v dnešních modelech nižší. Jedná se například o barometr, vlhkoměr nebo sensor tepu srdce.

3.2. Systémy k určení pozice

V případě složitější systémů a aplikací, které dovolují i pohyb v exteriéru, je často nezbytné znát přesnou aktuální pozici uživatele. Téměř synonymem pro systémy určení pozice je GPS, který je provozovaný Spojenými státy americkými.

Zařízení dnes ale často kombinují údaje z družic pro různé systémy, aby například dosáhly větší přesnosti nebo aby stále dokázaly fungovat v případně výpadku některého z nich. Jedná se o GLONASS, který provozuje Rusko, a Galileo spravované Evropskou unií.

Uvedené systémy jsou použitelné pouze v exteriéru s přímou viditelností na oblohu. Uvnitř budov je možné určovat pozici vůči bodům se známou pozicí – například Wi-Fi směrovače nebo systém iBeacon.

Tabulka 3. Srovnání specializovaných zařízení pro rozšířenou realitu [38, 39, 40, 44]

	Google Glass	Microsoft HoloLens	Lenovo New Glass C220	ZapBox
Cena	1500 USD	3000 USD	?	30 USD
Rozlišení obrazu	640 × 360	1280 × 720 (na jedno oko)	?	Záleží na vloženém telefonu
Datum představení	2013	Březen 2016	Leden 2018	2016
Samostatné	Ano	Ano	Ano	Ne
Bezdrátové	Ano	Ano	Ne	Ano
Hmotnost	36 gramů	579 gramů	60 gramů	?
Ovládání	Touchpad, hlas	Pohled, gesta, hlas	?	Gesty dodaným ovladačem

3.3. Specializovaná zařízení

Ačkoliv mobilní telefon může představovat dostačující zařízení při konzumaci obsahu rozšířené reality, tak mohou nastávat situace, kdy je například buď vyžadováno více výpočetního výkonu, nebo pohodlnější způsob ovládání. Jsou to situace, kde nacházejí využití specializovaná zařízení.

V následujících podkapitolách jsou podrobněji rozvedeny Google Glass, Microsoft HoloLens, Lenovo New Glass C220 a ZapBox. Výše uvedená tabulka (Tabulka 3) shrnuje jejich jednotlivé vlastnosti a umožňuje tak i jejich snadné porovnání.

3.3.1. Google Glass

Google Glass byly představeny v roce 2013. Jedná se o zařízení ve formě klasických průhledných brýlí, které se skládají z kamery, LCoS displeje a touchpadu pro ovládání. Ovládání je možné také pomocí hlasových příkazů. Dále obsahuje gyroskop, akcelerometr, magnetometr, senzor pro snímání okolního světla a senzor přiblížení. Celkem váží 36 gramů. [40]

Brýle dokáží fungovat buď samostatně, nebo ve spojení s mobilním telefonem. Typicky mohou sloužit například pro zobrazování notifikací na nově příchozí emaily a zprávy. Rozhraní brýlí je možné si představit jako sled karet, které krouží kolem hlavy a pomocí touchpadu je v nich možné listovat a aktivovat je. Dále je s nimi možné například telefonovat, posílat zprávy, fotit, vyhledávat, zobrazovat mapy a živě vysílat obraz z brýlí. [40]

3.3.2. Microsoft HoloLens

HoloLens je zařízení pro rozšířenou realitu od společnosti Microsoft. Uvedeno bylo v roce 2016. Vzhledem se liší od Google Glass. Prvky rozšířené reality jsou vnímány oběma očima a celkově je zařízené mnohem robustnější a rovnoměrně obepíná celou hlavu v úrovni očí. Z toho vyplývá také vyšší váha, kdy celé zařízení váží 579 gramů. Ovládání probíhá pohledem v kombinaci s gesty ruky. K dispozici je i ovládání hlasovými příkazy s možností komunikace s virtuální asistentkou Cortanou. Vývojáři aplikací mohou definovat i své vlastní hlasové příkazy. Zařízení také obsahuje gyroskop, akcelerometr, magnetometr a senzor pro snímání úrovně okolního osvětlení. [41]

Hlavním využitím zařízení je rozšiřování reálné scény trojrozměrnými objekty od jednotlivých předmětů až po celé komplikované scény či modely reálných míst. Pro brýle k dispozici aplikace „HoloStudio“, ve které je možné další objekty modelovat a následně i vytisknout na 3D tiskárně. Microsoft pro brýle také implementoval verzi Skype komunikátoru. Uživatel volající ze svého notebooku nebo mobilního

telefonu pak může sledovat prostředí, tak jak ho vnímá uživatel s nasazenými HoloLens. Kromě toho umožňují i spuštění 2D aplikací, které jsou vytvořené pro UWP (univerzální Windows platforma). [41]

3.3.3. Lenovo New Glass C220

Brýle od společnosti Lenovo se velmi podobají Google Glass. Rozšířená realita je shodně vnímána jedním okem, zatímco druhé je stále zaměřeno jen na reálný svět. Hlavním rozdílem je dodatečné zařízení, které se k brýlím připojuje, a možnost zvolit si, které oko bude zaměřeno na displej s rozšířenou realitou. [42]

Poprvé byly přestaveny v roce 2017 a jejich vylepšená verze v roce 2018. Hlavní předností by měla být kombinace s umělou inteligencí, která by měla být schopná identifikovat objekty v reálném světě. Brýle jsou určeny například pro zobrazování pokynů případně ke vzdálené pomoci při současném vysílání obrazu. [42]

3.3.4. ZapBox

Hlavní cílem ZapBoxu je poskytnout zařízení pro rozšířenou realitu za zlomek ceny stávajících zařízení, která často stojí i tisíce amerických dolarů. Hlavním rozdílem od předchozích uvedených zařízení je v tom, že se jedná pouze o pasivní rámeček, do kterého se vloží mobilní telefon, který se v kombinaci s nainstalovanou aplikací postará o vykreslování virtuálních prvků. S rámečkem je dodávána přídatná čočka na kameru telefonu, aby byl zajištěn reálnější pozorovací úhel. [43]

Hlavním nedostatkem je pravděpodobně nutnost osazení místnosti záchytnými body, které jsou aplikací využívány k lepší identifikaci prostředí. Ovládání probíhá gesty s pomocí dodaných ovladačů, které jsou také označeny záchytnými body, aby je aplikace dokázala lépe detekovat. [44]

3.3.5. Budoucnost zařízení pro rozšířenou realitu

Všechna představená zařízení mají jednu společnou vlastnost – vyžadují nošení externího hardwaru. V budoucnu může být tato slabina odstraněna pomocí kontaktních čoček. Základní myšlenka vychází z faktu, že obraz je vnímán pouze čočkou oka, takže vše kromě „displeje“ (zejména nějaká forma kamery, výpočetního čipu, uložení a komunikačního čipu) může být umístěno na části oka,

která nevnímá okolní obraz – typicky duhovka a její širší okolí. Energie pro pohon by musela být získávána přímo z těla, ať už z kinetické energie mrkání nebo chemickou reakcí se slzami. V ještě více vzdálené budoucnosti by dokonce i kontaktní čočky mohly být nahrazeny přímou úpravou informace přenášené z oka do mozku.

4. Využití webového prohlížeče

Před vývojem všech aplikací je nutné si nejen určit cílová zařízení, ale také konkrétní programovací jazyk a prostředí, pro které bude program vyvíjen. Vzhledem k tomu, že rozšířená realita od zařízení vyžaduje relativně hodně výpočetního výkonu, jehož potřebu dokáže uspokojit zejména grafická karta, tak je dobré volit takový jazyk a prostředí, kde lze jednoduše přistupovat ke zmíněnému hardwaru.

Dalším hlediskem je fakt, že na trhu existuje celá řada různých operačních systémů, takže vývoj specifické aplikace pro každou platformu by bylo zejména časově náročné, ale v delším horizontu i náročnější na údržbu a provádění případných aktualizací. To platí, i přestože přes 99,2 % (listopad 2017) [46] mobilních zařízení (mobilní telefony a tablety) využívá buď operační systém Android od Googlu nebo iOS od Applu, takže by stačilo pravděpodobně vyvinout aplikaci jen pro 2 platformy.

Kvůli těmto nevýhodám je lepším řešením sáhnout po webovém prohlížeči. Ve spojení JavaScriptu s WebGL a dalšími rozhraními, které jsou v JavaScriptu dostupné, je možné implementovat funkcionalitu rozšířené reality pouze pomocí zmíněných technologií a obsah uživatelům dodávat skrze webový prohlížeč. Ačkoliv je mezi uživateli využíván větší počet prohlížečů, respektive vykreslovaných jader, tak většina z nich si je tak hodně podobných, že není nutné vynakládat téměř žádné úsilí vytváření odlišných verzí jednoho programu. A i v okamžiku, kdy se v něčem některá vykreslovací jádra liší, tak pouze stačí vytvořit odlišnou větev chování a tudíž lze všem prohlížečům na všech operačních systémech dodávat jednu verzi programu.

Pro web je navíc momentálně vyvíjen standard WebXR, aby bylo možné snadno v rámci webové stránky pracovat s externími zařízeními.

4.1. *JavaScript*

JavaScript za poslední roky prošel velkým rozvojem a nyní nabízí mnoho bohatých rozhraní, které dovolují vyvíjet téměř plnohodnotné aplikace běžící ve webovém prohlížeči. Svůj díl na tom má zejména konkurence mezi výrobci prohlížečů, kteří

se snaží všechny novinky rychle implementovat do svých produktů. Samozřejmě bez standardizace by veškerá snaha vyšla vniveč, a tak výrobci prohlížečů často i spolupracují na jejím vytváření.

Široká škála funkcí ale sama o sobě není dostačující. Důležité také je, aby provádění kódu v JavaScriptu bylo co možná nejrychlejší. Moderní interprety splňují i tento požadavek. Díky zmíněným faktům je dnes možné programovat plnohodnotné a graficky výpočetně náročné aplikace, které jsou spouštěny jako webové stránky.

4.2. WebGL

WebGL (Web Graphics Library, česky webová grafická knihovna) je webový standard pro vykreslování dynamické trojrozměrné scény v JavaScriptu s využitím grafické karty. Je založeno na OpenGL ES, což přineslo výhodu snadné implementace v prohlížečích a do budoucna může WebGL snadno následovat novinky uvedené do OpenGL ES [47]. Důležitou vlastností je, že veškerá podpora je závislá pouze na webovém prohlížeči bez nutnosti instalovat další zásuvné moduly.

V současné době existují 2 verze WebGL. Oběma předcházely experimentální verze, které se v kódu získávají s příslušným prefixem. Experimentální první verze se v prohlížečích začaly objevovat už na přelomu let 2010 a 2011. Po dokončení specifikace se i implementace v prohlížečích koncem roku 2013 ustálily. V současné době se pracuje na specifikaci druhé verze, jejíž první finální verze byla vydána v dubnu 2017. I přesto hlavní prohlížeče na začátku roku 2017 již tuto verzi začaly plnohodnotně podporovat.

Podrobnější informace o implementaci jednotlivých verzí v hlavních prohlížečích lze získat z následující tabulky (Tabulka 4). V Safari je možné přístup k WebGL 2 získat pouze povolením experimentálních funkcí.

Tabulka 4. Podpora verzí WebGL (leden 2018) [48, 49]

	IE / Edge	Mozilla Firefox	Google Chrome	Safari
Experimentální WebGL	od verze 11 (říjen 2013)	od verze 4 (březen 2011)	od verze 8 (prosinec 2010)	od verze 5.1 (červenec 2011)
WebGL	-	od verze 24 (srpen 2013)	od verze 33 (únor 2014)	od verze 8 (říjen 2014)
Experimentální WebGL2	-	od verze 25 (září 2013)	-	-
WebGL2	-	od verze 51 (leden 2017)	od verze 56 (leden 2017)	-

WebGL kontext je objekt, přes který lze volat všechny dostupné funkce příslušného kontextu. Nutno podotknout, že pokus o získání více WebGL kontextů nad vykreslovacím plátnem není možné. Jakmile je získán jeden, tak volání o všechna další vrací hodnotu *null*. Způsob získání jednotlivých kontextů je předveden na následující ukázce. Pro každé volání *getContext* je vždy vytvořeno nové plátno. Uvedený kód je možné použít i k detekci podporovaných verzí v prohlížeči.

```

"use strict";

var gls = ["experimental-webgl", "webgl", "webgl2"];

for (let i = 0; i < gls.length; i++) {
    let canvas = document.createElement("canvas");
    let gl = canvas.getContext(gls[i]);
    // console.log(gls[i], ": ", gl);
    console.log(gls[i] + (gl ? " je " : " není ") + "podporováno.")
}

```

Ukázka kódu 1. Získání WebGL kontextu

4.2.1. WebGL 1

WebGL standard byl vytvořen v reakci na nedostačující možnosti pro vytváření graficky náročných scén ve webových prohlížečích. Základy vznikly kolem roku 2007 na základě experimentů s HTML5 elementem *canvas*. Původní specifikace elementu zahrnovala pouze 2D kontext, ale je možné implementovat další kontexty. Mozilla a Opera přišly s implementací 3D kontextu [50]. V roce 2009 se

projektu ujala společnost Khronos Group a specifikace WebGL 1 byla vydána v roce 2011. WebGL v první verzi vychází z OpenGL ES 2.

4.2.2. WebGL 2

WebGL ve své druhé verzi je založeno na OpenGL ES 3 a proti WebGL 1 přineslo několik nových funkcí. Není ale s touto verzí nutně zpětně kompatibilní, i když rozdíly by měly být minimální [51].

Mezi hlavní novinky patří například přímá podpora pro tzv. „Multiple render targets“ (česky „více cílů pro vykreslování“), které umožňují vykreslování do více bufferů nebo textur během jednoho vykreslovacího cyklu. Funkce byla k dispozici ve WebGL 1 pouze skrze rozšíření. Jedno z hlavních použití je technika nazvaná „deferred shading“ (česky „odložené stínování“), které dovoluje rychlé vykreslení scény s mnoha světelnými zdroji, protože složitost vykreslení pak nezávisí na počtu světel ale jen na celkovém počtu pixelů, kterou jsou osvětleny. [51]

Další novinkou je vykreslování instancovaných útvarů (anglicky „Instanced geometry drawing“). Díky tomu je možné zobrazit více instancí jednoho útvaru v rámci jednoho vykreslovacího cyklu a snížit tak zátěž CPU. Každá instance může mít své vlastní atributy (zpravidla transformační matice), takže je možné kreslit mnoho podobných objektů – například osoby v davu nebo stromy v lese. [51]

Další z novinek je například rozhraní *WebGLTransformFeedback*, které nebylo k dispozici ve WebGL 1 ani formou rozšíření. Umožňuje zachycení výstupu z vertex shaderu do bufferu, který je pak často využit jako vstup do dalšího vykreslovacího cyklu. Díky tomu data nikdy neopustí GPU a dochází k odlehčení zátěže CPU. Funkce nachází časté využití při implementaci částicových systémů. [51, 52]

Přidána byla také podpora pro 3D textury a bylo odstraněno omezení pro použití textur, které nemají rozměr druhé mocniny, takže je na ně nyní možné používat všechny funkce včetně například tzv. „mip-mappingu“. [51, 53]

4.2.3. Budoucnost akcelerované grafiky na webu

WebGL je velkým přínosem pro možnost využití 3D grafiky na webu a s příchodem verze 2 se její možnosti ještě dále rozšířily. Nově vznikající grafická API ale dnes

fungují na mnohem nižší úrovni abstrakce a díky snížené režii tedy dosahují lepšího výkonu než OpenGL a také WebGL, které z něho vychází. Jedná se zejména o Direct3D od Microsoftu, Metal od Applu a Vulkan od Khronos Group. Všechna jsou postavená na podobných konceptech, ale žádné z nich není dostupné pro všechny platformy. [47]

Proto se tým kolem vykreslovacího jádra WebKit rozhodl přijít s návrhem nového standardu, který by byl plně přizpůsobený použití na webu, zajišťoval co nejefektivnější přístup ke zdrojům grafické karty a byl dostupný pro všechny platformy. Dále by podle nich měl zajišťovat možnost využití obecných výpočtů na GPU a snadnou kooperaci s ostatními webovými standardy. [47]

WebGPU, jak je původní návrh pojmenován, se inicializuje podobně jako WebGL přes element `canvas` (viz Ukázka kódu 2). Na rozdíl od WebGL je ale více objektově orientované, díky čemuž je dosaženo vyšší efektivnosti. Namísto situace, kdy se stav nastavuje před každým vykreslením, tak s pomocí WebGPU se vytváří objekty, které stav samy reprezentují, takže je možné nezbytné validace provádět s předstihem už ve chvíli, kdy je stav vytvořen, čímž dochází ke snížení času nutného ke spuštění vykreslení [47]. WebGPU není momentálně podporováno žádným prohlížečem. Podporu lze nalézt pouze v technickém sestavení Safari a nočních sestaveních WebKitu [54].

```
let canvas = document.querySelector("canvas");  
let gpu = canvas.getContext("webgpu");
```

Ukázka kódu 2. Inicializace WebGPU, převzato z [47]

Pro budoucnost graficky na webu v kombinaci s virtuální a rozšířenou realitou je také důležitý vznikající standard WebXR (viz kapitola 4.5).

4.3. Údaje ze senzorů

Jak je popsáno v kapitole 3.1, používaná zařízení a zejména mobilní telefony disponují celou řadou různých sensorů, které dokáží poskytovat užitečné informace o stavu zařízení. V kapitole jsou podrobně popsány a vysvětleny funkce a události poskytující podporu pro získání údajů o aktuální orientaci a

momentálním pohybu zařízení. Údaje jsou poskytovány akcelerometrem a gyroskopem. Tyto sensory jsou v moderních mobilních telefonech velmi rozšířené, protože jsou nezbytně nutné pro automatické otáčení obrazovky v závislosti na poloze zařízení.

Následující tabulka shrnuje podporu funkcí pro zjištění orientace a pohybu zařízení v hlavních webových prohlížečích. Dostatečná podpora pro většinu aplikací byla dosažena už přibližně v roce 2013, kdy byla implementace doplněna do prohlížeče Internet Explorer. Řádek s částečnou podporou pouze znamená, že prohlížeče neimplementují událost pro kalibraci kompasu [55].

Tabulka 5. Podpora pro zjištění orientace a pohybu zařízení (leden 2018) [55]

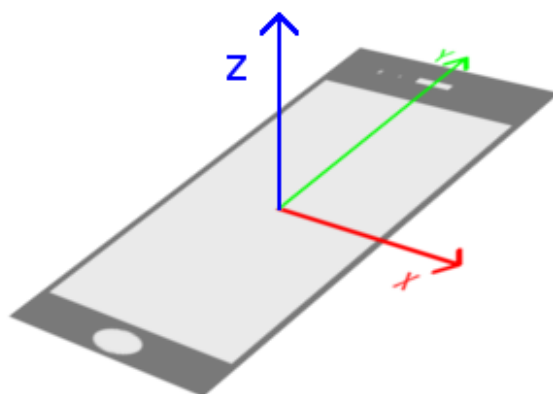
	IE / Edge	Mozilla Firefox	Google Chrome	Safari
Částečná podpora	od verze 11 (říjen 2013)	od verze 6 (srpen 2011)	od verze 7 (říjen 2010)	od verze 4.3 (březen 2011)
Plná podpora	od verze 12 (červenec 2015)	-	-	-

4.3.1. Orientace zařízení

Orientaci zařízení je možné zjišťovat pomocí události *deviceorientation*. Ta vrací objekt, který obsahuje hodnoty *alpha*, *beta* a *gamma*. *Alpha* představuje údaj z kompasu. Číslo v rozmezí 0-360 vyjadřuje úhel, který zařízení svírá s pomyslnou čarou vedoucí mezi magnetickými póly. Reprezentuje pohyb zařízení kolem osy Z, jak možné vidět v následující ilustraci (Obrázek 6).

Hodnota *beta* vyjadřuje v jakém prostorovém vztahu je vůči sobě vrch a dno telefonu – určuje pohyb zařízení kolem osy X (viz Obrázek 6). Bude-li vrch telefonu pozvednut výše než dno, tak bude hodnota *beta* stoupat až na 90 do chvíle, kdy telefon bude v plně vertikální poloze. Hodnota bude naopak přesně opačně klesat až k -90, bude-li dno telefonu výše. To platí v případě, že displej zařízení je vždy namířen vzhůru. Pokud bude telefon namířen displejem k zemi, tak se hodnoty budou pohybovat v rozmezí 90 až 180 případně -90 až -180, podle toho zda je výše vrch nebo dno telefonu.

Podobný způsobem se chová i hodnota γ , která reprezentuje pohyb zařízení kolem osy Y, jak ilustruje následující obrázek. Bude-li se levá strana telefonu zvedat nad pravou, tak hodnota γ poroste od 0 až na 90. V opačném případě, kdy se pravá strana bude zvedat nad levou, tak hodnota bude klesat od 0 až k -90. Pokud ale bude displej mířit směrem k zemi, tak se hodnoty obrátí, což znamená, že hodnoty od 0 do 90 budou v situaci, kdy je pravá strana nad levou a stejně tak pro hodnoty od 0 do -90 pro stav, kdy je levá strana nad pravou.



Obrázek 6. Ilustrace os rotace zařízení

Například zařízení ležící na vodorovném stole směřující svým vrchem k severu bude mít všechny hodnoty blízké se nule. Zařízení postavené kolmo ke stolu v tzv. „portrait“ pozici bude vracet hodnotu β 90 a γ 0. Změní-li se pozice zařízení do tzv. „landscape“ pozice s pravou stranou jdoucí dolů, tak se hodnoty β a γ prohodí. Zařízení ležící na vodorovném stole displejem směrem ke stolu bude mít hodnotu β 180 a hodnotu γ 0.

Tabulka 6. Shrnutí hodnot orientace zařízení

	Osa	Rozmezí hodnot
Alpha	Z	0 – 360
Beta	X	-180 – 180
Gamma	Y	-90 – 90

Následující ukázka demonstruje přiřazení události, získání výše popisovaných hodnot a jejich vypsání do konzole. Kromě toho také naznačuje možnost zobrazení upozornění v případě, že funkce podporovaná není – ať už zařízením nebo použitým prohlížečem.

```
if (window.DeviceOrientationEvent) {
    window.addEventListener("deviceorientation", function(e) {
        console.log(e.alpha, e.beta, e.gamma);
    });
} else {
    console.log("%cDeviceOrientation není podporováno.", 'color: red');
    window.alert("Toto zařízení pravděpodobně nemá všechny požadované senzory, aby zjistilo svou orientaci.");
}
```

Ukázka kódu 3. Přiřazení události pro zjištění orientace zařízení

4.3.2. Pohyb zařízení

Orientaci zařízení je možné zjišťovat pomocí události *devicemotion*. Funguje stejně jako událost pro získání orientace, ale liší se objektem, který je obsluhující funkcí vrácen jako parametr.

Parametr obsahuje 4 různé informace [56]:

- Akceleraci zařízení v m/s^2
- Akceleraci zařízení včetně vlivu gravitace v m/s^2
- Rychlost rotace zařízení ve stupních za sekundu
- Interval v milisekundách vyjadřující, jak často jsou data získávána přímo z hardwaru

Oba objekty s akcelerací obsahují hodnoty zrychlení pro každou z 3 os, jak jsou ilustrovány v obrázku výše (Obrázek 6). Objekt s akcelerací včetně vlivu gravitace zpravidla nenabízí tolik využití, ale může být použit v případě, že zařízení není schopno odstranit vliv gravitace z výsledných dat, což může zpravidla nastat na zařízeních, která nemají gyroskop [57].

Objekt s rychlostí rotace obsahuje podobně jako objekt s orientací 3 hodnoty: *alpha*, *beta* a *gamma*. Každá z nich odpovídá rychlosti rotace podle jedné osy (viz Obrázek 6): *alpha* podle osy X, *beta* podle osy Y a *gamma* podle osy Z.

Následující ukázka demonstruje přiřazení události, získání všech výše popisovaných hodnot a jejich vypsání do konzole s popisky a jednotkami. Kromě toho také naznačuje možnost zobrazení upozornění v případě, že událost podporovaná není – ať už zařízením nebo použitým prohlížečem.

```
if (window.DeviceMotionEvent) {
    window.addEventListener("devicemotion", function(e) {
        console.log("Akcelerace (m/s^2)");
        console.log(e.acceleration.x, e.acceleration.y,
e.acceleration.z);
        console.log("Akcelerace s vlivem gravitace (m/s^2)");
        console.log(e.accelerationIncludingGravity.x,
e.accelerationIncludingGravity.y, e.accelerationIncludingGravity.z);
        console.log("Rychlost rotace (stupně za sekundu)");
        console.log(e.rotationRate.alpha, e.rotationRate.beta,
e.rotationRate.gamma);
        console.log("Interval získávání dat (ms)")
        console.log(e.interval);
    });
} else {
    console.log("%cDeviceMotion není podporováno.", 'color: red');
    window.alert("Toto zařízení pravděpodobně nemá všechny požadované senzory, aby zjistilo svůj relativní pohyb.");
}
```

Ukázka kódu 4. Přiřazení události pro zjištění pohybu zařízení

4.4. Vstup z kamery

Přístup ke kameře je v prohlížeči získán pomocí javascriptové funkce `MediaDevices.getUserMedia()`. Ta jako parametr přijímá objekt se specifikací požadovaných zdrojů. Typicky se jedná o žádost o audio a video s možnými dalšími upřesněními. V případě videa se například může jednat o upřesnění preference, zda má být získán přístup k přední či zadní kameře má-li zařízení k dispozici obě – což mobilní telefony zpravidla mívají. Dále je možné upřesnit preferované rozlišení obrazu z kamery, je ale také možné si vyžádat přesné hodnoty, případně využít atributů `min` a `max` a tím si vyžádat rozmezí, které je nutné pro správné fungování aplikace.

Kód (Ukázka kódu 5) demonstruje způsob získání přístupu k zadní kameře zařízení s preferovaným rozlišením 1280×720. Přístup k mikrofonu vyžádán není. V ukázce nejsou využity upřesňující atributy pro požadované rozlišení obrazu,

protože během testování s tím některé prohlížeče měly problém a přístup ke kameře zamítaly.

V ukázce je také speciálně řešeno zachytávání chyb. V zavislosti na typu chyby je možné řešení větvit a nabízet tak uživateli konstruktivní řešení v případě, že je možné chybu vyřešit. Pokud je rozlišení kamery příliš nízké, není možné z pohledu uživatele problém nijak řešit za předpokladu, že aplikace by s nižším rozlišením nedokázala pracovat. Takový problém by však bylo možné vyřešit na pozadí bez nutnosti informovat uživatele. Další chybou, která může nastat, je například požadovaný přístup ke kameře, která je již v danou chvíli využívána jinou aplikací. V takovém případě může uživatel jinou aplikaci vypnout, aby bylo možné přístup ke kameře zajistit.

```
var video = document.querySelector("video");

navigator.mediaDevices.getUserMedia({
  // přístup k mikrofonu není vyžadován
  audio: false,
  // přístup ke kameře je požadován
  // preference je vyjádřena pro zadní kameru, je-li k dispozici,
  // a pro rozlišení 1280×720
  video: { width: 1280, height: 720, facingMode: "environment" }
}).then(function(stream) {
  // přiřazení streamu z kamery
  video.srcObject = stream;
}).catch(function(error) {
  // odchyčení případných chyb
  let msg = "";
  if (error.name === "ConstraintNotSatisfiedError" || error.name ===
  "NotFoundError") {
    msg = "Your camera does not meet required resolution for this
  application to work.";
  } else if (error.name === "TrackStartError") {
    msg = "Your camera is probably used by different application
  at this time.";
  } else {
    msg = "There was an unknown error when accessing your
  camera.";
  }
  alert(msg);
  console.log("%cVideo Error: " + error, 'color: red');
  console.log(msg);
});
```

Ukázka kódu 5. Získání přístupu ke kameře v JavaScriptu

V závislosti na typu a nastavení prohlížeče je před povolením přístupu zpravidla zobrazena žádost o svolení. Je to důležité zejména kvůli tomu, že přístup ke kameře, který by byl prováděn bez vědomí uživatele, by mohl představovat vážné riziko pro soukromí. Některé prohlížeče uživateli umožňují udělovat trvalé povolení, takže při dalším přístupu na stejnou stránku už žádost zobrazena není. Přístup ke kameře může také být zakázán pro celý prohlížeč v závislosti na tom, jaký operační systém zařízení využívá.

V prohlížečích založených na projektu Chromium (Google Chrome, Opera) je navíc možné funkci `MediaDevices.getUserMedia()` spouštět pouze v případě, že je webová stránka načtena přes zabezpečený kanál – https. Týká se to všech tzv. „mocných“ funkcí, které pracují například s platebními údaji, dostávají přístup k sensorům nebo mají vyšší stupeň kontroly nad zařízením. [58]

Podpora popisované funkce je začátkem roku 2018 již dostatečná, jak lze vysledovat v následující tabulce. Ve stejné tabulce je také možné vidět, že plnou podporu nabízí všechny hlavní prohlížeče už více než rok.

Tabulka 7. Podpora funkce `getUserMedia()` (leden 2018) [59]

	IE / Edge	Mozilla Firefox	Google Chrome	Safari
Částečná podpora	-	od verze 17 (říjen 2012)	od verze 21 (červen 2012)	-
Plná podpora	od verze 12 (červenec 2015)	od verze 42 (listopad 2015)	od verze 53 (srpen 2016)	od verze 11 (září 2017)

4.5. Virtuální a rozšířená realita v prohlížeči

Podporu pro virtuální a rozšířenou realitu ve webových prohlížečích se aktuálně snaží definovat 2 různé standardy – WebVR a WebXR. První zmíněný pokrývá pouze virtuální realitu. WebXR se snaží být následovníkem WebVR a mělo by nabízet obecnou podporu pro všechny reality.

4.5.1. WebVR

WebVR je javascriptové API, které si klade za cíl zpřístupnit využití virtuální reality ve webovém prohlížeči. Definuje možnosti zpřístupnění zařízení pro virtuální realitu webovým aplikacím, které pak mohou využívat informace o pohybu v rámci

prostorové scény. Základní myšlenou je usnadnění vývoje aplikací využívajících virtuální realitu a také zjednodušení jejího využití bez závislosti na tom, jaké zařízení je použito. [60, 61]

Tabulka 8. Podpora WebVR (leden 2018) [62]

	Edge	Mozilla Firefox	Google Chrome	Samsung Internet
Podpora	od verze 15 (duben 2017)	od verze 54 (červen 2017)	od verze 57 (březen 2017)	od verze 57 (duben 2016)
Omezení	Pouze „Windows Mixed Reality“	Pouze Windows	Pouze přes experimentální funkce. Pouze některá VR zařízení.	Pouze pro telefony Samsung Galaxy. Pouze některá VR zařízení.

Aktuální podpora v prohlížečích je stále pouze částečná, často dostupná pouze skrze zapnutí experimentálních funkcí. Edge v aktuální verzi 16 podporuje pouze „Windows Mixed Reality“. Firefox poskytuje podporu pouze pro operační systém Windows. Podpora pro iOS je dostupná pouze v jeho nočních sestaveních. Chrome od verze 57 obsahuje podporu pouze v rámci experimentálních funkcí. Navíc podporuje jen Oculus Rift a HTC Vive na zařízeních s Windows. Údaje jsou shrnuty v tabulce výše (Tabulka 8). [62]

Tabulka 9. Podpora VR zařízení v prohlížečích pro hlavní OS (leden 2018) [63]

	Oculus Rift	HTC Vive	Windows MR	Cardboard	Daydream	Samsung Gear VR
Firefox	pouze Windows	pouze Windows a macOS	-	-	-	-
Edge	-	-	pouze Windows	-	-	-
Chrome	pouze Windows	pouze Windows	-	pouze Android	pouze Android	-
Samsung Internet	-	-	-	pouze Android	-	pouze Android

Předchozí tabulka (Tabulka 9) uvádí podrobněji podporu pro jednotlivá VR zařízení. Zahrnuty jsou všechny webové prohlížeče, které podporují alespoň jedno zařízení. Operační systém Linux momentálně není podporován žádným zařízením.

Následující ukázka kódu (Ukázka kódu 6) demonstruje způsob přístupu k zařízení virtuální reality v JavaScriptu v rámci WebVR.

```
if (navigator.getVRDisplays) {
    console.log("WebVR je podporováno");
    // získání VR zařízení, která jsou připojena
    navigator.getVRDisplays().then(function(displays) {
        // pokud je displej k dispozici, tak je použit k zobrazení
        if (displays.length > 0) {
            vrDisplay = displays[0];
            // přiřazení vrDisplay - nyní ho lze libovolně používat
        }
    });
}
```

Ukázka kódu 6. Získání VR zařízení, převzato z [64]

4.5.2. WebXR

WebVR je zaměřené pouze na virtuální realitu, ale opomíjí rostoucí zájem o rozšířenou realitu. Proto v říjnu 2017 Mozilla navrhla, aby začal být rozpracovaný standard WebVR nahrazován novým standardem WebXR, jehož úmyslem je soustředit se na všechny různé reality – včetně virtuální i rozšířené. WebXR se stále drží stejné myšlenky, kterou je zjednodušení vývoje webových aplikací využívajících smíšených realit v rámci všech platforem. [65]

Cíle WebXR se dají shrnout do následujících 4 bodů [66]:

- Umožnit detekci připojených zařízení pro virtuální i rozšířenou realitu
- Umožnit dotazování se na schopnosti připojených zařízení
- Umožnit získávání polohy a orientace zařízení
- Umožnit zobrazování obrazu na zařízení v dostatečné obnovovací frekvenci

Kromě typického využití WebXR pro tvorbu her, zejména v kombinaci s WebGL, by další aplikací WebXR by mohlo být například 360° a 3D video. Klasická videa jsou dnes na webu všudypřítomná a typickým prvkem, který je doprovází, je možnost

přepnutí na celou obrazovku. Stejně tak by mohlo fungovat tlačítko „Zobrazit ve VR“, které by zajistilo přepnutí přehrávání do vnějšího zařízení. [66]

Dalším potenciálním využitím je 3D vizualizace. Webové stránky ji mohou nabízet jako doplněk ke klasickému vykreslení na monitoru. Prohlížení 3D modelů, architektonických návrhů nebo i prostá vizualizace dat mohou být efektivnější a přesvědčivější při použití virtuální nebo rozšířené reality. Například aplikace nabízející nábytek a bytové doplňky mohou virtuální realitu využít jako formu prezentace nabízeného zboží, díky čemuž si zákazník může mnohem lépe představit výsledný vzhled zvoleného zboží v kontextu dané místnosti. [66]

Následující ukázka kódu (Ukázka kódu 7) demonstruje způsob přístupu k zařízení v JavaScriptu v rámci WebXR.

```
navigator.xr.requestDevice().then(device => {
  if (device) {
    onXRAvailable(device);
  }
}, err => {
  console.error("Nepodařilo se získat XR zařízení: ", err);
});
```

Ukázka kódu 7. Návrh způsobu získání XR zařízení v rámci WebXR, převzato z [66]

5. Návrh a implementace ukázkových úloh

5.1. *Popis cíle*

Cílem ukázkových úloh je demonstrovat využití dat ze senzorů a dalších vstupů, které mobilní zařízení poskytují. Dále jsou testovány možnosti interakce v rámci jednoduchého systému rozšířené reality implementovaném na mobilním zařízení. Byly navrženy celkem 3 různé úlohy a každá představuje odlišný přístup k využití získávaných údajů. Implementace kromě úloh samotných zahrnuje i několik pomocných knihoven pro usnadnění práce jak s WebGL, tak například s webovými sokety.

Navržen je koncept rozšířené reality tvořený systémem komunikujícího mobilního a desktopového zařízení. Klíčovými prvky systému jsou vizualizace scény a interakce s uživatelem. Obraz generovaný na počítači je snímán kamerou v mobilním telefonu, kde proběhne detekce změny pohybu, a zjištěné údaje jsou předány po síti. Data obdrží počítač, který je aplikuje a vygeneruje nový obraz. Důležitý je fakt, že nový obraz na desktopovém počítači je generován v závislosti na způsobu interakce uživatele s mobilním telefonem.

V následujícím textu jsou nejprve podrobně popsány navržené způsoby detekce změny polohy a orientace, které jsou klíčové pro ovládání všech aplikací. Dále je vysvětlen způsob komunikace mezi mobilním a desktopovým zařízením. Důležitá je i schopnost komunikace více mobilních zařízení s jedním počítačem, díky čemuž je možné implementovat například hry pro více hráčů.

5.2. *Detekce posunutí*

Všechny implementované úlohy vyžadují interakci s uživatelem ve formě zadávání směru dalšího pohybu. Detekce posunutí pomocí akcelerometru se ukázala být problematická, protože senzor slouží zejména k poskytování údajů o relativní změně a získávání absolutních jednotek se projevilo jako obtížné a velmi nepřesné i přes fakt, že přímo objekt vrácený událostí obsahuje kromě dat samotných i interval, vůči kterému se zrychlení vztahuje. Z údajů bylo možné spočítat absolutní

vzdálenost pohybu, ale přesnost se i v nejlepší situaci pohybovala pouze v jednotkách centimetrů, častěji však i v desítkách centimetrů.

Navíc zařízení, na kterém byl sensor testován, pravděpodobně obsahovalo chybu přímo v hardwaru. Ta mohla být způsobena i předchozím pádem zařízení, který mohl způsobit odchýlení sensoru. Situace jen poukázala na to, že se nelze na sensor spolehnout, protože jiná zařízení mohou vykazovat i značně vyšší chybovost, kterou nelze dopředu predikovat.

Kvůli všem těmto faktům bylo nutné se přiklonit ke složitější a výpočetně náročnější detekci změny pozice identifikačního znaku (tzv. markeru) z obrazu kamery pomocí WebGL, u které se však lze spolehnout na jednoznačné chování napříč zařízeními.

Primárně je využíváno WebGL 2 v kombinaci s rozšířením *EXT_color_buffer_float*. Pokud není WebGL 2 podporováno, tak se využije WebGL 1 s rozšířením *OES_texture_float* nebo *OES_texture_half_float*. Ve všech případech se jedná o rozšíření, která umožňují číst informace z textury ve *float* formátu.

```
// target, level, internalformat, width, height, border, format, type,
// ArrayBufferView? pixels)
// WebGL 1, OES_texture_float
gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA, 320, 180, 0, gl.RGBA, gl.FLOAT,
null);
// WebGL 1, OES_texture_half_float
gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA, 320, 180, 0, gl.RGBA,
floatExtension.HALF_FLOAT_OES, null);
// WebGL 2, EXT_color_buffer_float
gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA32F, 320, 180, 0, gl.RGBA, gl.FLOAT,
null);
```

Ukázka kódu 8. Využití rozšíření při specifikaci textury

Postup detekce je možné rozdělit do následujících kroků:

- Aktualizace textury podle obrazu z kamery
- První průchod vykreslovacím řetězcem pro detekci markeru
- Druhý průchod vykreslovacím řetězcem pro detekci markeru a redukci obrazu
- Přečtení dat a nalezení markeru

Aktualizace dat textury je velmi snadná, protože obrazový zdroj z kamery je napojen na HTML element `video`, který je možné přímo použít jako parametr v příslušné funkci `gl.texImage2D`. Poté následuje napojení datových bufferů a tzv. framebufferu, který zajistí, že se nebude vykreslovat do plátna ale do textury. Pro oba průchody vykreslovacím řetězcem je použit stejný vertex shader v kombinaci s rozdílnými fragment shadery, protože veškerá implementace probíhá právě v nich. Vertex shader slouží jen pro přesun souřadnic textury do fragment shaderů.

Při prvním průběhu vykreslovacím řetězcem dojde ke kontrole všech pixelů původního obrazu na přítomnost zelených pixelů. Zároveň se kontroluje i červená a modrá složka, které musí mít co nejnižší hodnotu, aby se vyloučila detekce bílé barvy. Dále jsou pixely pro zjednodušení nazývány pouze jako „zelené“ pixely. Následující obrázek ilustruje chování první části. Pixely jsou kontrolovány po skupinách 16 (4×4) pixelů, aby se dosáhlo zmenšení výstupního obrazu. V každé skupině se sečte počet zelených pixelů a do každého výstupního pixelu ve skupině jsou zapsána totožná data, kde první číslo (tzv. červená složka) je počet zelených pixelů ve skupině, druhé číslo (tzv. zelená složka) je původní X souřadnice pixelu, který je v levém horním rohu skupiny, a třetí číslo (tzv. modrá složka) je původní Y souřadnice pixelu, který je v levém horním rohu skupiny. Jsou dva důvody, proč jsou do 16 různých pixelů zapisována stejná data. Prvním je paralelní zpracování, kdy neexistuje způsob, jak mít pod kontrolou pořadí zpracování pixelů. Druhým důvodem je různé chování prohlížečů při zmenšování výstupního obrazu, protože výstupní textura má čtvrtinové rozměry a různé prohlížeče vybírají pixely z rozdílných pozic do výstupu. Na výkonu se to nijak znatelně neprojevuje a ziskem je sjednocené chování napříč všemi možnými prohlížeči. V obrázku (Obrázek 7) je z důvodu čitelnosti ukázán zápis pouze prvního pixelu ve skupině.

Y\X	12	13	14	15	16	17	18	19	20	21	22	23
36	4,12,36				5,16,36				0,20,36			
37												
38												
39												
40	8,12,40				15,16,40				0,20,40			
41												
42												
43												
44	0,12,44				3,16,44				2,20,44			
45												
46												
47												

Obrázek 7. Ilustrace první části detekce markeru z obrazu kamery

Po doběhnutí první části je ihned spuštěna druhá část. Jejím vstupem je výstup z první části. Průběh je podobný. Liší jen ve způsobu výpočtu výsledných pixelů. Pracuje po skupinách 9 pixelů (3×3), jak ilustruje následující obrázek (Obrázek 8). Pokud jsou celkově nalezeny méně než dva zelené pixely, tak dojde k vyřazení skupiny. V každé validní skupině se pak spočítá vážený aritmetický průměr vůči počtu nalezených zelených pixelů a souřadnicím. Pro předchozí obrázek by vyšla výsledná souřadnice $X = 14,92$ a $Y = 39,57$ pro 37 zelených pixelů. Tyto hodnoty jsou zapsány na všech 9 pozic zpracovávané skupiny, ze stejných důvodů, které jsou uvedeny výše při vysvětlování prvního kroku.

Y\X	3			4			5			POČET	X	Y
	r	g	b	r	g	b	r	g	b			
9	4	12	36	5	16	36	0	20	36	37	552	1464
10	8	12	40	15	16	40	0	20	40		14,919	39,568
11	0	12	44	3	16	44	2	20	44			

Obrázek 8. Ilustrace druhé části detekce markeru z obrazu kamery

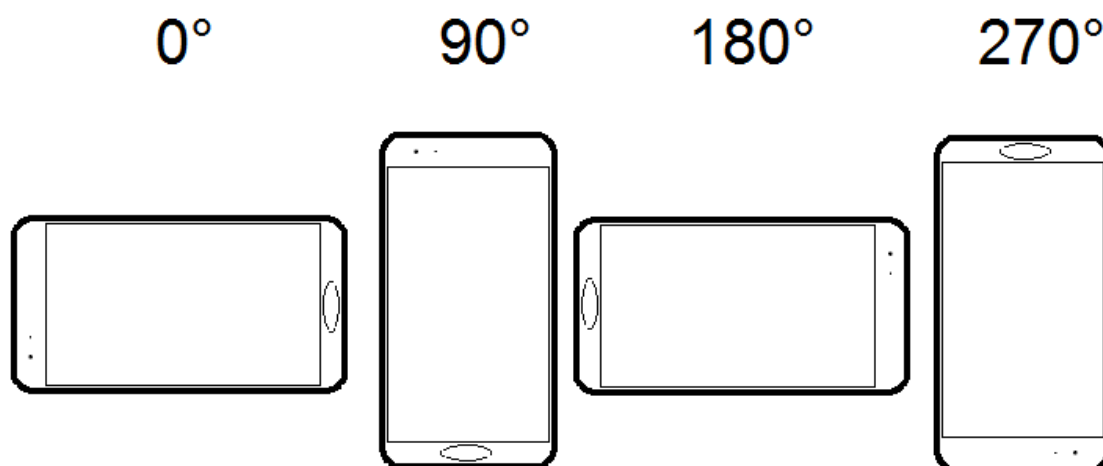
Vzhledem k tomu, že docházelo dvakrát ke zmenšování obrazu, tak výsledné rozměry výstupní textury jsou 144krát menší než vstupní obraz z kamery. Hlavní důvod snahy o co největší zmenšení je v tom, že čtení dat z textury do JavaScriptu

zabírá přibližně 90 % celkového času průběhu celého algoritmu a čas narůstá s množstvím ponechaných výsledných pixelů.

Posledním krokem je finální přečtení pixelů do připraveného `Float32Array` pole v JavaScriptu a jejich projití pro nalezení oblasti s největším počtem zelených pixelů. Jako výsledek jsou prezentovány vypočtené průměrné souřadnice z druhého kroku.

5.3. Detekce orientace

K získání orientace zařízení je využívána událost `DeviceOrientationEvent`, jejíž parametry a hodnoty jsou podrobněji popsány v kapitole 4.3.1. Hodnoty ze senzoru je nutné před použitím zpracovat. Senzor vrací data velmi často, takže se vždy zpracovává průměr ze dvou po sobě jdoucích údajů. Dále jsou data upravována, aby bylo možné jak ve vertikální tak v horizontální pozici zjistit natočení zařízení. Nejprve se v závislosti na hodnotě `gamma` upraví hodnota `beta` – pokud je `gamma` kladné číslo, tak se od 180 odečte hodnota `beta`. Pokud je po této operaci hodnota stupňů záporná, tak se přičte 360 stupňů. Danou posloupností výpočtů se získá hodnota v intervalu $< 0; 360$ stupňů, s nulou v situaci, kdy vrch zařízení míří vlevo. Výsledné hodnoty pro konkrétní natočení jsou uvedeny v následujícím obrázku.



Obrázek 9. Zpracované údaje ze senzoru pro provádění rotace

Spouštěcí hodnota pro provedení rotace je v aplikacích nastavená od výchozí pozice na 20 stupňů s rozmezím rotace 15 stupňů. To například znamená, že

rotace vpravo by se prováděla pro hodnoty $\langle 110^\circ; 125^\circ \rangle$, kdy je telefon blízko vertikální pozici s vrškem mířícím nahoru.

5.4. Způsob komunikace

Vzhledem k povaze aplikace je nutné, aby komunikace probíhala v reálném čase. K tomuto účelu jsou využita technologie tzv. webových soketů. S jejich pomocí je možné odesílat data ze serveru ke klientovi, aniž by se klient musel periodicky dotazovat na server, zda pro něj má nějaká nová data. Namísto toho klient pouze otevře se serverem trvalé spojení a naslouchá na příslušné události. Veškerá činnost pak už závisí na serveru, který při přijetí dat automaticky vše přeposílá na zaregistrované klienty.

Tabulka 10. Podpora webových soketů (leden 2018) [67]

	IE / Edge	Mozilla Firefox	Google Chrome	Safari
Plná podpora	od verze 10 (zaří 2012)	od verze 11 (leden 2012)	od verze 16 (říjen 2011)	od verze 7 (říjen 2013)

K využití webových soketů je nutná jejich podpora i na serveru, aby se klientský program mohl k jejich příjmu zaregistrovat. Zejména kvůli výslednému zjednodušení, kdy nedochází ke tříštění mezi více programovacích jazyků, byl serverový kód napsán také v JavaScriptu s využitím Node.js a modulu „ws“, který zajišťuje podporu pro webové sokety.

Byl proveden i pokus s tzv. „Push API“, se kterým však bylo spojeno několik problémů. Ačkoliv komunikace probíhala v reálném čase, tak zpoždění v některých webových prohlížečích bylo v rozpětí 500 – 1000 milisekund, což jsou naprosto nepřijatelné hodnoty pro nerušený a hladký uživatelský zážitek. Po bližším prozkoumání bylo zjištěno, že je to dáno tím, že Push API je určeno pro situace, kdy je zpoždění do 1 sekundy stále akceptovatelné – například pro chatovací webové aplikace jako jsou WhatsApp nebo Facebook Messenger nebo pro e-mailové webové aplikace. Díky této technologii mohou zobrazovat upozornění na nově příchozí zprávy i ve chvíli, kdy záložka s danou aplikací není aktivní. Jedná se o situace, ve kterých nevádí, že se zpoždění pohybuje ve stovkách milisekund, protože uživatel ve většině případů prodlevu kolem 1 sekundy pravděpodobně ani

nezaregistruje. Dalším problémem byla podpora v prohlížečích. V době provádění pokusu (březen 2017) pouze prohlížeče založené na jádrech Gecko (zejména Mozilla Firefox) a Chromium (zejména Google Chrome a Opera) disponovali podporou.

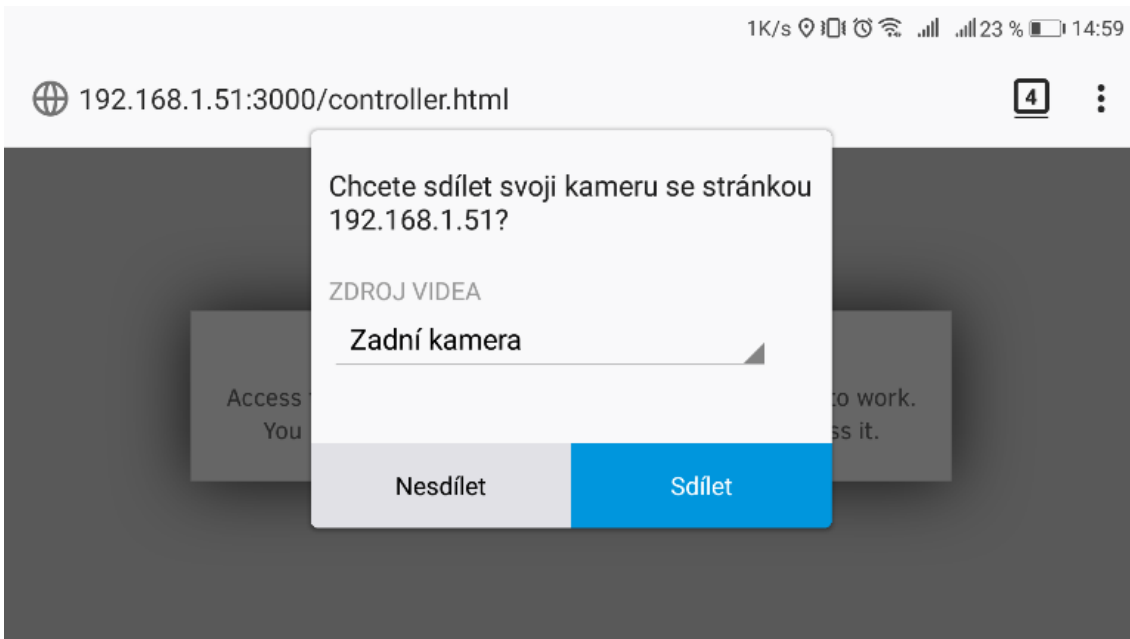
5.5. Struktura úloh

Základním principem všech úloh je rozdělení na 2 spolupracující části, které spolu jednosměrně komunikují skrze server. Obecně lze první část všech úloh nazvat jako kontrolér, protože slouží k ovládání. Mobilní telefon či tablet jsou typickými zařízeními pro tuto roli. Druhou částí je přijímač, který dostává data od kontroléru a modifikuje podle nich scénu. Přijímačem může být libovolné zařízení, ale zpravidla by se mělo jednat o notebook nebo stolní počítač. Teoreticky je ale možné použít jakékoliv zařízení, které splňuje základní požadavky – připojení k internetu a webový prohlížeč podporující WebGL.

5.5.1. Kontrolér

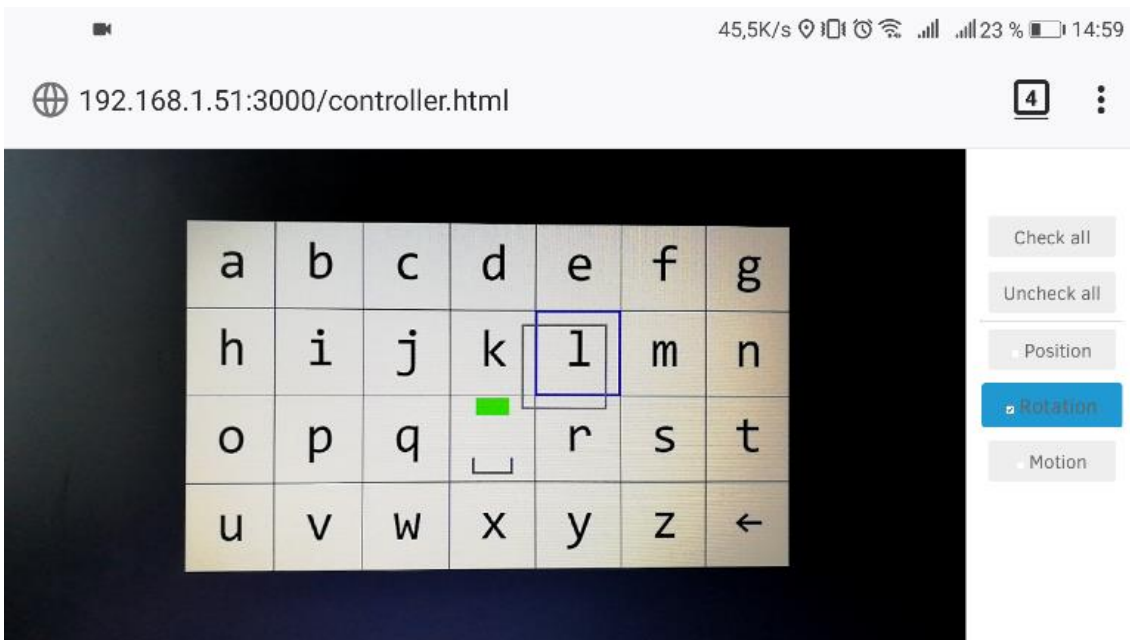
Kontrolér slouží k ovládání scény v přijímači. Používá k tomu údaje o rotaci ze senzorů a údaje o pohybu získávané detekcí markeru z obrazu kamery. Podrobné informace o implementaci detekce markeru v obraze lze nalézt v kapitole 5.2. Implementace získání dat o rotaci je podrobně uvedena v kapitole 5.3.

Před načtením kontroléru je uživatel požádán o povolení přístupu ke kameře (viz Obrázek 10). Poté se zobrazí základní uživatelské rozhraní s obrazem z kamery v levé části stránky a s ovládacími prvky na pravé straně (viz Obrázek 11). V případě obrácené orientace stránky se bloky přeskupí tak, aby obraz z kamery byl nahoře a ovládací prvky dole. Je toho docíleno v CSS pomocí tzv. flexboxů.



Obrázek 10. Žádost o povolení přístupu ke kameře v prohlížeči

Pomocí jednotlivých přepínačů je možné zapínat a vypínat funkce pro detekce změny pozice, orientace a relativního pohybu. Všechny je možné kombinovat a jejich využití zpravidla závisí na konkrétní aplikaci.



Obrázek 11. Grafické uživatelské rozhraní kontroléru

5.5.2. Přijímač

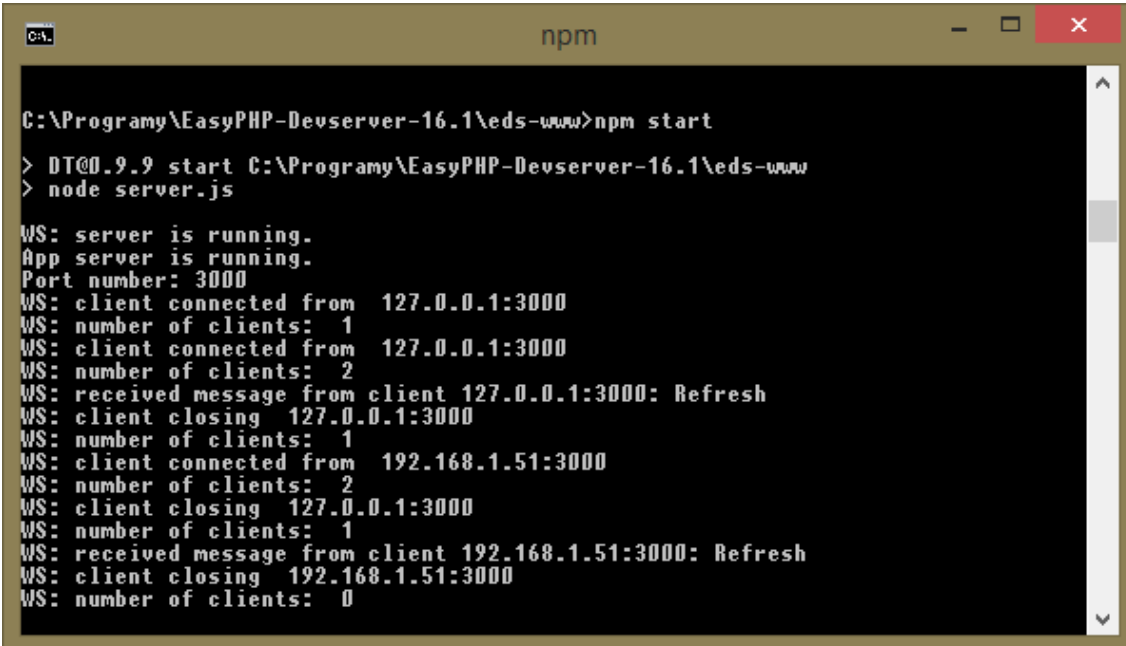
Přijímač je pouze pasivní destinací všech dat od kontrolérů. Podoba jednotlivých přijímačů se může velmi lišit, protože s přijatými daty je možné provádět velmi různorodé operace. Více informací o různých formách přijímačů je možné nalézt v následující kapitole 5.6.

5.5.3. Server

Server je naimplementován v JavaScriptu pomocí Node.js prostředí. Základním požadavkem na server byla možnost pracovat s webovými sokety. Funkcionalitu zajišťuje balíček „ws“. Server dokáže zpracovávat data od více zdrojů a předávat je všem připojeným klientům. Základní zpřístupnění zdrojů poskytuje kombinace balíčků „express“ a „serve-static“. Aby dále bylo možné snadno přijímat data ve formátu JSON, tak je využit balíček „body-parser“.

Server funguje na velmi jednoduchém principu. Na nastavené adrese čeká na příjem dat, která jsou zpracována přiřazenou funkcí, která jen potvrdí jejich přijetí a zajistí jejich přeposlání všem klientům.

Následující obrázek (Obrázek 12) je snímek okna konzole s běžícím serverem, kde jsou prezentovány základní výpisy o připojených klientech.

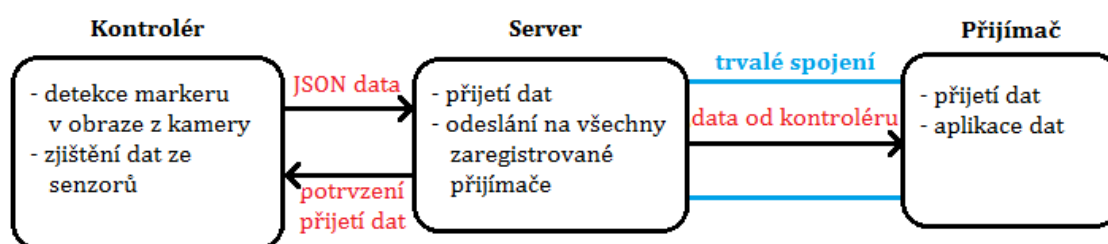


```
npm
C:\Programy\EasyPHP-Devserver-16.1\eds-www>npm start
> DT@0.9.9 start C:\Programy\EasyPHP-Devserver-16.1\eds-www
> node server.js
WS: server is running.
App server is running.
Port number: 3000
WS: client connected from 127.0.0.1:3000
WS: number of clients: 1
WS: client connected from 127.0.0.1:3000
WS: number of clients: 2
WS: received message from client 127.0.0.1:3000: Refresh
WS: client closing 127.0.0.1:3000
WS: number of clients: 1
WS: client connected from 192.168.1.51:3000
WS: number of clients: 2
WS: client closing 127.0.0.1:3000
WS: number of clients: 1
WS: received message from client 192.168.1.51:3000: Refresh
WS: client closing 192.168.1.51:3000
WS: number of clients: 0
```

Obrázek 12. Výpis z konzole serveru

5.5.4. Komunikace mezi kontrolérem a přijímačem

Obrázek 13 názorně ilustruje, jak probíhá komunikace mezi jednotlivými částmi aplikace. V rámci kontroléru se provede detekce markeru a zjištění požadovaných údajů ze senzorů. Všechna data jsou v JSON formátu odeslána na server, který je v rámci trvale navázaného spojení předá dále na přijímač.



Obrázek 13. Grafické znázornění komunikace mezi jednotlivými částmi

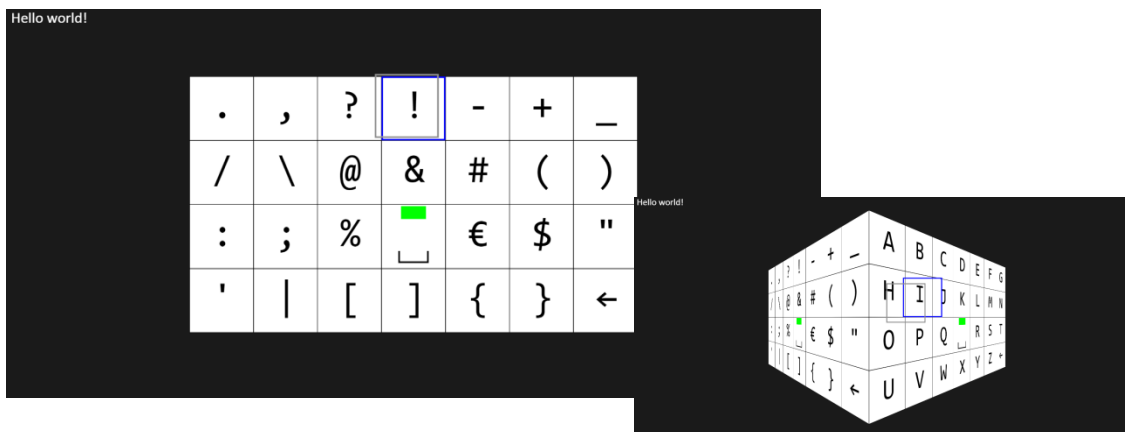
Komunikace nemusí nutně probíhat v podobě 1 kontrolér a 1 přijímač. Je možné využít i více kontrolérů, pokud na to bude přijímač připraven a dokáže s přijatými daty naložit. Data jednotlivých kontrolérů jsou vždy rozlišitelná připojeným identifikátorem. Jedním kontrolérem je případně možné ovládat i více různých přijímačů. Server data žádným způsobem nerozlišuje a vše přeposílá.

5.6. Úlohy

Byly navrženy celkem 3 různé přijímače a každý představuje odlišný přístup k využití dodávaných údajů od kontroléru.

5.6.1. Abeceda

Úloha „Abeceda“ si klade za cíl umožnit psaní textu pouze s ovladačem (mobilním telefonem) v ruce. Dvě hlavní funkcionality zahrnují výběr znaku na aktuální obrazovce a rotaci mezi 4 obrazovkami se znaky. Od kontroléru vyžaduje data o pozici a rotaci.

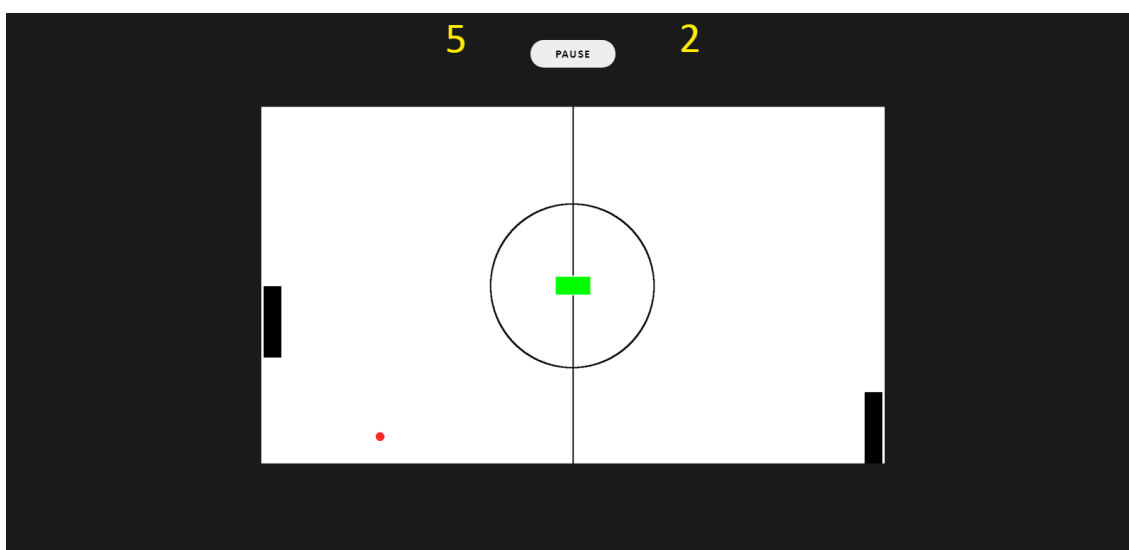


Obrázek 14. Ukázka úlohy Abeceda

5.6.2. Pong

Principem hry pong je obelstít soupeře a dostat červenou „kuličku“ přes jeho černý herní kámen. Jeho ovládání je zajištěno kontrolérem s pomocí detekce pozice markeru. Úloha je navíc zjednodušená v tom, že stačí detekovat pouze vertikální změnu pozice.

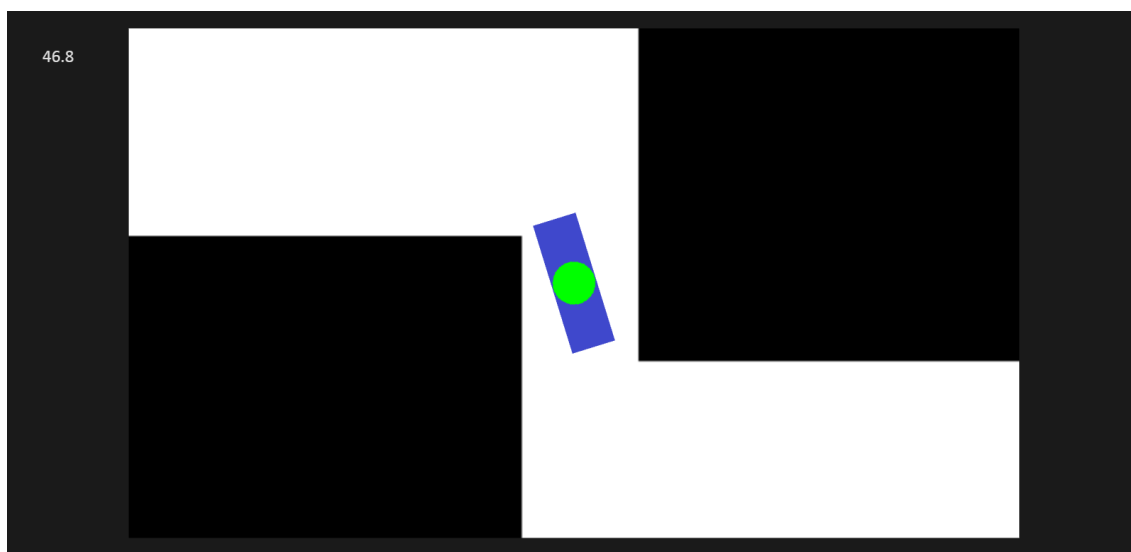
Hlavní princip, který úloha prezentuje, spočívá v něčem jiném. Hra poskytuje 2 různé módy. Prvním je hra pro jednoho hráče, kdy je druhý hráč simulován počítačem. Je ale také možné hrát ve dvou hráčích pomocí dvou kontrolérů, což představuje hlavní cíl demonstrace této úlohy. Obecně je možné v úlohách využívat i více kontrolérů, pokud na to úloha bude přizpůsobena.



Obrázek 15. Ukázka úlohy Pong

5.6.3. Bludiště

Bludiště je hra pro jednoho hráče. Cílem hry je přenést modrý kámen se zeleným markerem do červeně označeného cíle. Struktura bludiště je dána vstupním obrázkem, ze kterého je zobrazován jen výřez, aby byl zajištěn průchod bludištěm. Po bílých místech je možné se volně pohybovat. Překážky mají černou barvu. Kromě posunu je možné herní kámen i otáčet, aby se například vešel do úzkých míst. Z kontroléru se tedy využívají údaje o změně pozice a rotace. Navíc dochází i k měření času, který byl potřebný pro přesun ze startu do cíle.

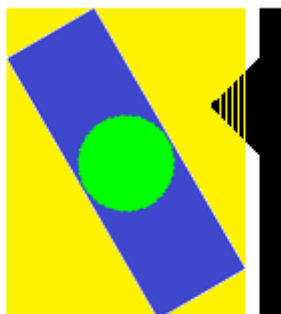


Obrázek 16. Ukázka úlohy Bludiště

Aby bylo možné řešit kolize s okolními překážkami a zároveň dovolit rotaci, tak bylo nutné vymyslet algoritmus, který by dokázal kolize vyhledat v obdélníkových obrazových datech. Jediný způsob, jak získat barvu pozadí obrázku v plátně, je pomocí funkce `getImageData`. Ta však pracuje pouze s obdélníkovými oblastmi a překážky mohou být i vyčnívající, jak je ilustrováno na situaci na následujícím obrázku (Obrázek 17), kde žluto-černě prolínající se oblast by běžně byla označena jako kolize, ale je nutné ji ignorovat, protože herní kámen se na označeném místě nenachází.

Pro detekci dojde k získání celé obdélníkové oblasti, která pokrývá celý herní kámen (modrá a zelená část na následujícím obrázku). Pro každý bod je pak provedena kontrola, zda se nachází v jednom ze 4 žlutých trojúhelníků. Pokud se

v něm nachází, tak je možné pokračovat, protože herní kámen se v této lokaci nenachází. Pokud ale v žádném trojúhelníku není, tak je nutné provést test barvy pozadí, protože kontrolovaný pixel se nachází pod herním kamenem.



Obrázek 17. Vysvětlení detekce kolizí při rotaci v úloze bludiště

5.7. Pomocné knihovny

Pomocné knihovny zahrnují pět souborů, které obsahují často využívané funkce, které jsou v ukázkách používány.

5.7.1. transforms3d.js

Soubor „transforms3d.js“ obsahuje skupinu objektů a funkcí umožňujících používat v JavaScriptu vektorové a maticové výpočty, které jsou pro práci s počítačovou grafikou nezbytné. Dále obsahuje funkcionalitu pro práci s kvaterniony, kubickými křivkami a bikubickými plochami a speciální objekt pro práci s kamerou ve scéně. Založen je na balíčku Java tříd nazvaném „transforms“ používaném v rámci předmětů PGRF na FIM UHK.

Knihovna byla vyvinuta již při implementaci bakalářské práce [68]. Oproti původní verzi 1.0 bylo v nové verzi 1.1 opraveno několik nedostatků. Prvním z nich je porovnávání hodnot pouze dvojitým znakem „=“, které zajišťuje pouze prosté porovnání namísto trojitého „==“, které zajistí striktní porovnání i typu porovnávaných hodnot. Dále byla pro přiřazení lokálních proměnných používána konstrukce „var“ namísto správnější konstrukce „let“. Nakonec byly upraveny konstruktory, aby byla zajištěna práce pouze s číselnými hodnotami a všechny nevalidní hodnoty vracely korektní výjimku. Pro knihovnu byly také dopsány testy.

5.7.2. **webgl-utils.js**

Soubor „webgl-utils.js“ obsahuje objekt *Utils*, který implementuje několik funkcí, které nacházejí časté uplatnění při práci s WebGL. Patří mezi ně například funkce pro inicializaci WebGL, překreslovací smyčky a shaderů, převod matic a vektorů na *Float32Array* a zjednodušuje práci při generování scény.

Knihovna byla vyvinuta již při implementaci bakalářské práce [68] a oproti původní verzi 1.0 bylo v nové verzi 2.0 provedeno mnoho úprav a doplnění nových funkcí. V rámci generování scény bylo doplněno generování normál a souřadnic do textur a také opraveno generování indexů pro kvádr. Pro generování stěny byla doplněna podpora pro změnu rotace stěny pomocí rotační matice. Stejně jako v případě knihovny „transforms3d.js“ byly také opraveny nedostatky při porovnání hodnot pouze dvojitým znakem „=“ a používání konstrukce „var“. V mnoha funkcích byly odstraněny již nepotřebné varianty s prefixy, protože podpora v prohlížečích se již ustálila. Pro knihovnu byly také dopsány testy.

Vzhledem k použitým syntaktickým konstrukcím je podpora zajištěna pouze v prohlížečích Edge 12 a vyšší, Chrome 45 a vyšší, Firefox 22 a vyšší, Opera 32 a vyšší a Safari 10 a vyšší.

5.7.3. **math2.js**

Soubor „math2.js“ obsahuje objekt *Math2*, který implementuje pokročilé matematické funkce, které nejsou v klasickém matematickém javascriptovém objektu *Math* obsaženy.

Knihovna vychází z implementace bakalářské práce [68] a v nové verzi byla opravena funkce *clamp*, protože pro záporné minimum a maximum nedocházelo ke korektnímu oříznutí hodnoty. Dále byla upravena funkce *sign*, aby fungovala identicky jako její doplněný protějšek v objektu *Math*.

5.7.4. **WS.js**

Soubor „ws.js“ zapouzdřuje použití webových soketů. Je využíván všemi příjemci pro příjem dat ze serveru. Základní funkce *setWebSocket* přijímá 3 parametry – funkce pro spuštění při otevření spojení, při příjmu nových dat a při uzavření

spojení. Funkce volaná při otevření spojení je bez parametru. Uživatelská funkce pro příjem nových dat dostává jako parametr objekt získaný z přijatých dat v JSON formátu. Funkce pro spuštění při uzavření spojení je bez parametru.

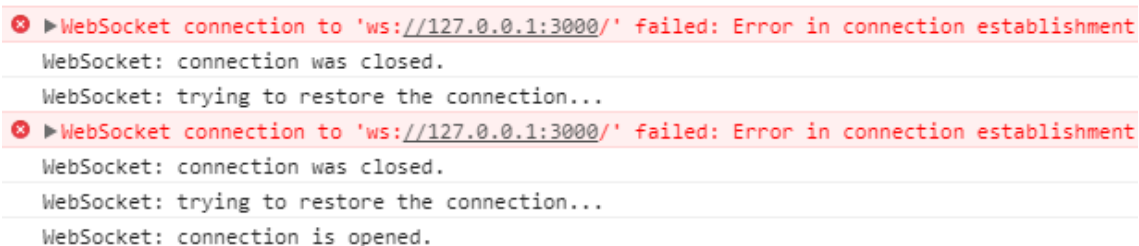
Aby nedocházelo k uzavření neaktivního spojení, tak je nastaven časovač, který každých 30 sekund odešle obnovovací prázdnou zprávu na server. Dále pokud k uzavření spojení dojde (například kvůli restartu serveru), tak automaticky dojde k pokusu o obnovení spojení (viz Obrázek 18).

Celá knihovna pro obsluhu webových soketů funguje samostatně bez potřeby dalšího kódu a je možné ji využít v libovolných situacích.

Následující ukázka demonstruje, jak vypadá implementace hlavních událostí v rámci popisované knihovny.

```
// událost otevření spojení
wsClient.onopen = () => {
    console.log("WebSocket: connection is opened.");
    open(); // uživatelská funkce
};
// událost pro přijetí nových dat
wsClient.onmessage = (e) => {
    let data = JSON.parse(e.data);
    message(data); // uživatelská funkce
};
// událost uzavření spojení
wsClient.onclose = () => {
    console.log("WebSocket: connection was closed.");
    console.log("WebSocket: trying to restore the connection...");
    clearInterval(refreshInterval);
    close(); // uživatelská funkce
    WS.setWebSocket(open, message, close); // restart
};
```

Ukázka kódu 9. Implementace hlavních události pro obsluhu webových soketů



```
✖ ▶WebSocket connection to 'ws://127.0.0.1:3000/' failed: Error in connection establishment
WebSocket: connection was closed.
WebSocket: trying to restore the connection...
✖ ▶WebSocket connection to 'ws://127.0.0.1:3000/' failed: Error in connection establishment
WebSocket: connection was closed.
WebSocket: trying to restore the connection...
WebSocket: connection is opened.
```

Obrázek 18. Výpis z konzole prohlížeče při pokusu o obnovení WS spojení

5.7.5. sender.js

Soubor „sender.js“ sjednocuje odesílání údajů na server v rámci kontroléru. Dá se chápat jako protějšek k souboru „ws.js“, který je popsán v předchozí kapitole a který odeslaná data přijímá. Vyčlenění do zvláštního souboru bylo provedeno, protože množství odesílaných údajů je velké a ke spuštění docházelo často i s rozdílem jednotek milisekund, čímž docházelo ke zbytečnému otevírání spojení.

Objekt obsahuje pouze jednu veřejnou funkci `add`, která slouží k přidávání dat do pole s ostatními daty. V rámci objektu je spuštěn časovač, který každých 10 milisekund odešle všechna nashromážděná data. 10 milisekund je interval, během kterého nedojde ke znatelnému zpoždění odeslání dat a zároveň se zamezí příliš častému odesílání.

Navíc při načtení souboru dojde k vygenerování identifikátoru, který slouží k rozlišení zpráv, pokud příjemce očekává data od více kontrolérů. K jeho připojení dochází ve funkci `add`, čímž se zajistí, že všechna odeslaná data budou identifikátorem označena.

```
setInterval(function() { // spuštění odesílacího intervalu
    if (data.length > 0) send();
}, 10);
function send() { // privátní funkce pro odeslání dat na server
    let request = new XMLHttpRequest();
    request.open('POST', '/ajax/data');
    request.setRequestHeader("Content-Type",
"application/json;charset=UTF-8");
    request.send(JSON.stringify(data));
    data.length = 0;
}
```

Ukázka kódu 10. Implementace odesílání dat

5.8. Dostupnost ukázek

Veškeré navržené a implementované ukázky jsou k dispozici na webové adrese <https://still-garden-74128.herokuapp.com/>. Krátké video všech aplikací je k dispozici v příloze práce a na serveru YouTube: <https://youtu.be/IFiF5DKtxV4>. Zdrojové kódy jsou také dostupné jak v příloze práce, tak v repositáři na GitHubu: <https://github.com/milankostak/Diploma-thesis>.

6. Výsledky a hodnocení

Na základě zkušeností s návrhem a implementací úloh bylo provedeno hodnocení jak z pohledu programátora tvořící aplikace pro rozšířenou realitu, tak z pohledu uživatele, který chce vytvořenou aplikaci používat. Dalším hodnotícím hlediskem je rychlost odezvy přes webové sokety. Zhodnocena je také dostupnost senzorů a vhodnost použití JavaScriptu pro implementaci. Dále je hodnocení zaměřeno na rychlost detekce markeru pomocí kamery.

6.1. Dostupnost rozšířené reality

Dostupnost rozšířené reality z hlediska koncového uživatele je dnes velmi pohodlná. Zařízení byla v minulosti často těžkopádná a objemná a volný pohyb s nimi byl obtížný. Dnes naopak ke spouštění aplikací stačí buď běžné dostupný mobilní telefon případně specializovanější zařízení jakým je například Google Glass. Všechny ale spojuje fakt, že jsou lehké, malé a jejich používání je pohodlné.

Možnosti vývoje aplikací, které využívají principů rozšířené reality, jsou dnes také již velmi bohaté. Pro obyčejné aplikace se jako dostačující ukázalo využití webového prohlížeče v kombinaci s JavaScriptem a WebGL. Pro specializovaná zařízení stačí zpravidla vyvinout aplikace pro operační systém, na kterém zařízení běží. Často se jedná o Android, v případě Microsoft HoloLens je pak možné spouštět všechny aplikace vyvinuté pro UWP (univerzální Windows platforma). Ve všech případech se ale jedná o běžné postupy, takže i z hlediska vývojáře aplikací je přístup k rozšířené realitě snadný.

6.2. Vhodnost použití JavaScriptu

Důvodem k použití JavaScriptu pro implementaci rozšířené reality bylo několik. V první řadě se jedná o jazyk, který podporují všechny hlavní webové prohlížeče, což činí implementaci aplikací značně jednodušší. JavaScript je navíc jazyk, v jehož prostředí je možné přistupovat k výkonu grafické karty pomocí standardu WebGL. Dále je rychlost provádění již na takové úrovni, že pouze velmi náročné aplikace vyžadující běh na mnoha jádrech procesoru by nebylo možné implementovat. Nakonec množství funkcí, které jsou dnes v JavaScriptu podporované, je velmi

široké včetně funkcí pro přístup k sensorům, kameře a také externím zařízením virtuální a rozšířené reality.

Díky použití JavaScriptu také není nutné řešit instalaci aplikací či dalších běhových prostředí, protože o veškerou funkcionalitu se stará webový prohlížeč. Ten je navíc používán denně téměř každým uživatelem počítače či mobilního telefonu, takže představuje i pohodlný způsob přístupu k aplikaci pro koncové uživatele. Z toho těží i možnost pro použití více mobilních telefonů pro například hry více hráčů. Jak je zmíněno, mobilní telefon vlastní téměř každý, takže spustit hru s přáteli je velmi snadné oproti nutnosti použití specializovaných zařízení, které nejsou tak rozšířené a jejichž cena je často velmi vysoká.

Jediným možným problémem je chybějící podpora využitých funkcí v některých prohlížečích. Zejména se jedná o starší verze, které ale dnes již nejsou příliš používány.

Celkově je JavaScript hodnocen jako dobrá volba a podařilo se splnit veškerá očekávání. Jediným nedostatkem vůči nativní aplikaci je nemožnost zabránit uzamknutí obrazovky, pokud s ní uživatel nijak neinteraguje. I tento problém ale bude v budoucnu odstraněn, protože W3C připravuje standard „Wake Lock API“, díky kterému bude možné požadovaného chování docílit.

6.3. Rychlost detekce markeru z obrazu

Jak je vysvětleno v kapitole 5.2, detekce posunutí probíhá pomocí detekce markeru v obraze kamery. Navržený algoritmus je poměrně komplikovaný, a proto bylo prováděno průběžné testování rychlosti jeho provádění, aby byly odhaleny případné problematické úseky.

Konečné testování bylo provedeno na třech různých zařízeních. Prvním z nich byl notebook s 2 jádrovým procesorem Intel Core i5 z generace Arrandale, grafickou kartou ATI Radeon HD 6300M a operačním systémem Windows 8.1. Druhým je mobilní telefon Honor 9 s čipem HiSilicon Kirin 960 s 8 jádrovým CPU Cortex A-73, Mali-G71 MP8 GPU a operačním systémem Android 7. Třetím testovaným zařízením je mobilní telefon Lumia 640 reprezentující starší generaci telefonů. Poháněn je čipem Snapdragon 400 s 4 jádrovým CPU Cortex-A7 a Adreno 305 GPU.

Nainstalovaným operačním systémem je Windows 10 Mobile ve verzi „Creators Update“.

Měření je provedeno pro 5 různých částí průběhu detekce. Prvním je naplnění bufferů. Druhou a třetí částí jsou jednotlivé průběhy vykreslovacím řetězcem. Čtvrtou částí je přečtení dat z grafické karty. Jedná se nejkritičtější část, protože zabere většinu stráveného času detekce. Poslední částí je samotné projití dat a nalezení oblasti, kde se marker nachází. Všechna uvedená měření jsou průměrem pro 1000 průběhů.

Tabulka 11. Rychlost jednotlivých částí detekce markeru v obraze, 1. část: naplnění bufferů, 2. a 3. část: jednotlivé průběhy vykreslovacím řetězcem, 4. část: přečtení dat z grafické karty, 5. část: projití přečtených dat

	1. část	2. část	3. část	4. část	5. část	Celkem	Procento času 4. části
Notebook	0,04 ms	0,11 ms	0,05 ms	8,96 ms	0,07 ms	9,23 ms	97,1 %
Telefon Honor 9	0,06 ms	0,46 ms	0,39 ms	3,36 ms	0,13 ms	4,40 ms	76,4 %
Telefon Lumia 640	0,16 ms	1,16 ms	1,28 ms	27,14 ms	0,54 ms	30,28 ms	89,6 %

Z tabulky je možné sledovat, že nejvíce času trvá čtení dat z grafické karty, protože se jedná o velké množství dat a navíc je nutná synchronizace činnosti procesoru a grafické karty, aby data mohla být zkopírována z paměti grafické karty do operační paměti. V této části byla měřena pouze jediná operace - `gl.readPixels`. Poslední sloupec tabulky pro zajímavost uvádí procento času stráveného touto operací v porovnání se zbytkem celého procesu, a jak je možné vidět, jedná se o úzké hrdlo celého algoritmu.

Jediným způsobem, jak 4. část zrychlit, je snížit množství dat, které by byly kopírovány z paměti grafické karty. Zároveň ale se snižováním množství dat dochází ke snížení celkové rozlišovací schopnosti algoritmu pro nalezení markeru. Bylo tedy nutné najít kompromis mezi zmíněnými 2 protichůdnými požadavky. Hlavním cílem bylo držet se v rámci 60 průběhů za sekundu, tj. 16,67 sekundy na

jeden cyklus detekce. To se nedařilo pouze na starším telefonu s méně výkonným čipem, kde bylo dosaženo pouze přibližně 33 průběhů za sekundu. Ale i přesto bylo možné dané zařízení použít jako kontrolér, kde by pouze velmi všímavý uživatel dokázal zpozorovat menší zpomalení reakcí.

Výše uvedená tabulka obsahuje měření pro stav, který byl zvolen jako ideální, kdy detekce markeru je stále velmi spolehlivá a rychlost detekce je stále akceptovatelná. Vstupní rozlišení obrazu z kamery bylo zvoleno na 1280×720 pixelů a algoritmus obraz nejprve zmenší 4 krát a v dalším průběhu ještě 3 krát. Výsledkem je obraz 107×60 pixelů, který obsahuje 6 420 pixelů.

Před zvolením ideálního stavu bylo provedeno testování s různými stupni zmenšování – „3 a 3“ (11 360 výsledných pixelů), „4 a 4“ (3 600 výsledných pixelů), obě kombinace „3 a 4“ a „4 a 3“ (6 420 výsledných pixelů), „5 a 3“ (4 128 výsledných pixelů) a „5 a 4“ (2 340 výsledných pixelů). Měření v prvních třech částech se v uvedených situacích nelišilo, pouze v druhé části byla v některých případech zaznamenaná odchylka 0,02 milisekundy. Rozdíly byly naměřeny pouze ve čtvrté a páté části. Pozornost je upřena hlavně na čtvrtou část, protože rozdíly v páté části byly také jen v setinách milisekundy.

Základem pro testování byla zvolena výchozí implementace „4 a 3“. Při použití zmenšování „4 a 4“ došlo ke zrychlení pouze o 1 %, což je zanedbatelné v porovnání s tím, že dojde ke snížení rozlišení detekce. Verze „3 a 4“ byla při dané operaci pomalejší o 2,9 %, i když měření by mělo být stejné. Důvod se nepodařilo odhalit. Verze „3 a 3“ byla pomalejší o 9,5 %, což při naměřených hodnotách znamená už téměř celou milisekundu. Pro verze „5 a 3“ a „5 a 4“ bylo očekávané rychlejší přečtení dat, ale i při několikrát zopakovaném měření docházelo k pomalejšímu čtení o 4 % respektive 2,6 % ve druhém případě. Přesná příčina není známa, ale je pravděpodobné, že limitu je dosaženo z důvodu nutnosti synchronizace činnosti CPU a GPU, jak je popsáno výše, která trvá vždy stejnou dobu, a čtení menšího množství dat se na výsledném čase projeví jen zanedbatelně. Kromě verze „3 a 3“ se měření ve všech ostatních situacích pohybovaly v rámci 0,44 milisekundy. To vše byly důvody, proč byla zvolena verze „4 a 3“ pro finální implementaci – dosahuje dobrého výkonu při dobrém rozlišení výsledné detekce.

Veškeré výše uvedené informace o měření rozdílů mezi jednotlivými způsoby zmenšování výstupní textury při detekci markeru jsou uvedeny v následující tabulce.

Tabulka 12. Shrnutí testování s různými stupni zmenšování při detekci markeru

	Počet pixelů	Čas pro přečtení dat z grafické karty	Procentulání rozdíl vůči výchozímu „4 a 3“
„3 a 3“	11 360	9,81 ms	+ 9,5 %
„3 a 4“	6 420	9,22 ms	+ 2,9 %
„4 a 3“	6 420	8,96 ms	-
„4 a 4“	3 600	8,88 ms	- 0,9 %
„5 a 3“	4 128	9,32 ms	+ 4,0 %
„5 a 4“	2 340	9,19 ms	+ 2,6 %

6.4. Dostupnost senzorů

Na dostupnosti senzorů stojí celý návrh ukázkových úloh. Naštěstí se ukázalo, že potřebné sensory (zejména akcelerometr a gyroskop) jsou v mobilních telefonech široce zastoupené. Zároveň i jejich přesnost a rychlost dodávání dat jsou postačující. Také podpora funkcí v JavaScriptu je již velmi rozšířená ve všech novějších verzích prohlížečů. Celkově je dostupnost senzorů hodnocena jako plně dostačující.

6.5. Rychlost odezvy

Rychlost odezvy komunikace závisí na několika faktorech. V první řadě je důležité, aby všechna zařízení měla kvalitní připojení k internetu. Pokud by některé ze zařízení bylo připojeno například k bezdrátové síti s nekvalitním signálem z důvodu velké vzdálenosti od zdroje, tak rychlost odezvy se může stát až nepoužitelnou. Množství odesílaných dat od kontroléru se pohybuje mezi 30-60 kB za sekundu, což je možné bez problému přenášet i přes 3G síť. Přesné množství se odvíjí od toho, kolik různých údajů je z kontroléru posíláno. V případě hry více hráčů (kontrolérů) se může přijímač stát úzkým hrdlem v případě, že není připojen ke spolehlivé síti.

Druhým důležitým faktorem je fyzické umístění stroje, na kterém je spuštěn server, protože veškerá data přes něj musí projít. Pokud je umístěn v rámci lokální sítě, tak rychlost odezvy je zpravidla natolik rychlá, že uživatelé nemají šanci postřehnout žádné zpoždění reakcí. Podle testů je průměrná doba nutná pro přenos dat od kontroléru přes server na přijímač 18,2 milisekund (průměr pro 100 měření). Všechny naměřené hodnoty jsou podrobně uvedeny v následující tabulce, kde jsou údaje rozděleny do sloupců podle částí programu. První sloupec uvádí dobu pro přenos dat z kontroléru na přijímač. Druhý sloupec vyjadřuje dobu nutnou pro přenos dat z kontroléru na server pomocí AJAX technologie. Třetí sloupec je pravděpodobně ten nejzajímavější, protože uvádí dobu potřebnou pro přenos dat ze serveru na přijímač, což je místo, kde se používají webové sokety. V případě sloupce, který uvádí hodnoty pro přenos z kontroléru na server, je nutné také brát v potaz fakt, že v krajním případě mohou být data zpožděna až od 10 milisekund z důvodu použitého způsobu odesílání dat, který je popsán v kapitole 5.7.5.

Tabulka 13. Doba potřebná pro přenos dat z kontroléru na přijímač při použití lokálního serveru

	Z kontroléru na přijímač	Z kontroléru na server	Ze serveru na přijímač
Průměr	18,2 ms	8,5 ms	9,7 ms
Medián	17 ms	8 ms	9 ms
Minimum	8 ms	5 ms	2 ms
Maximum	50 ms	35 ms	22 ms

Pro další potřeby testování reálnějšího použití byl využit hosting „Heroku“, kde je možné zdarma provozovat javascriptové aplikace. V rámci nastavení je možné si pouze zvolit region, ve kterém aplikace poběží. Při zvolení regionu Evropy se serverová část aplikace podle IP adresy nacházela v Irsku. I přes to je možné v následující tabulce (Tabulka 14) sledovat, že komunikace se serverem, který je vzdálen až několik tisíc kilometrů, je stále dobrá. Vzhledem k tomu, že server je spuštěn na jiném zařízení než kontrolér a přijímač, tak není možné přesně změřit čas nutný pro přenos dat na server a z něj, protože čas na něm se liší a zaručit přesnou synchronizaci na milisekundy nebylo možné. Průměrný čas pro 100

měření činil 52,0 milisekund, což jsou stále hodnoty, které jsou pro uživatele nepostřehnutelné. Vzhledem k tomu, že minimum se pohybovalo pouze 6 milisekund pod průměrným časem, tak lze usoudit, že čas mezi jednotlivými měřeními byl velmi stabilní. Pouze 5 naměřených hodnot přesáhlo 60 milisekund.

Tabulka 14. Doba potřebná pro přenos dat z kontroléru na přijímač při použití vzdáleného serveru

Průměr	52,0 ms
Medián	51 ms
Minimum	46 ms
Maximum	99 ms

7. Závěr

V práci byly představeny principy rozšířené reality a její vývoj od vzniku po současnost. Dále byly vysvětleny různé druhy přístupů k rozšířené realitě a uvedeny rozdíly vůči virtuálně realitě. Vysvětlen je pojem rozšířená virtualita a uvedeno je šest různých kategorií aplikací rozšířené reality včetně konkrétních příkladů a možností budoucích aplikací.

V další kapitole je uveden základní hardware nezbytný pro aplikace rozšířené reality. Jsou popsány používané druhy senzorů a dále jsou uvedeny příklady některých specializovaných zařízení v čele s Google Glass a Microsoft HoloLens. Jednou z nejdůležitějších vlastností se ukazuje být volnost pohybu. To je důvodem, proč jen bezdrátová zařízení dávají smysl pro použití v rámci rozšířené reality.

V oblasti softwarových technologií je v práci pozornost věnována využití webového prohlížeče v kombinaci s implementací aplikací v JavaScriptu. Ukázalo se, že JavaScript poskytuje dostatečné možnosti pro implementaci rozšířené reality. Prozkoumány byly možnosti získání údajů ze senzorů a obrazu z kamery. Vše je spojeno pomocí standardu WebGL, který umožňuje využívat hardwarově akcelerovanou grafiku přímo ve webovém prohlížeči.

Možnosti rozšířené reality a virtuality ve webovém prohlížeči byly podrobně vyzkoušeny v rámci navržených testovacích aplikací. Důraz byl kladen na využití údajů ze senzorů, obrazu kamery a WebGL. Byl navržen a odzkoušen koncept rozšířené reality využívající desktopového a mobilního zařízení umožňující interakci ve scéně. Implementované úlohy demonstrují využití dostupných dat v různých situacích. Základním principem všech úloh je rozdělení na kontrolér, který představuje mobilní zařízení, jehož údaje ze senzorů a kamery jsou načítány, zpracovány a odeslány, a přijímač, který data přijímá a různým způsobem aplikuje.

Naměřené hodnoty odezvy a komunikace se serverem se při použití technologií webových soketů a AJAXu prokázaly jako plně dostačující pro hladký průchod informací. Z hlediska uživatele zpravidla nedochází ke znatelnému zpoždění posílaných dat. Navržený způsob komunikace umožňuje i kooperaci více mobilních telefonů s jedním přijímačem.

Funkčnost testových úloh prokázala, že navržený koncept využití mobilních zařízení pro online zpracování snímaného videa a detekci pozice poskytuje dostatečně rychlou odezvu pro implementaci systémů rozšířené reality. Interakce v rámci jednotlivých úloh probíhaly bez znatelné prodlevy a intuitivně pouze pohyby mobilního telefonu.

Další směr výzkumu by se mohl zaměřit na zdokonalování navrženého algoritmu detekce markeru a způsobů komunikace mezi jednotlivými částmi. Potenciálním problémem je použití aplikace v síti s nekvalitním připojením k internetu. V dané situaci může docházet k zasekávání a dalšímu nežádoucímu chování. Řešením by byla například implementace predikce dalšího vývoje příchozích dat v situaci, kdy data náhle přestanou být přijímána. Problémy působí také funkce pro automatické vypínání displeje zařízení, pokud s ním neprobíhá žádná interakce. V rámci nativních aplikací je možné chování předejít, ale ve webových aplikacích neexistuje v současnosti žádný způsob, jak situaci předcházet. Větší pozornost by také mohla být věnována připravovanému standardu WebXR, pomocí kterého je možné ve webovém prohlížeči využívat externích zařízení pro rozšířenou i virtuální realitu.

8. Reference

1. AZUMA, R., Y. BAILLOT, R. BEHRINGER, S. FEINER, S. JULIER a B. MACINTYRE. Recent advances in augmented reality. *IEEE Computer Graphics and Applications* [online]. **21**(6), 34-47 [cit. 2017-03-26]. DOI: 10.1109/38.963459. ISSN 02721716. Dostupné z: <http://ieeexplore.ieee.org/document/963459/>
2. MILGRAM, Paul a Fumio KISHINO. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*. 1994, **77**(12), 1321-1329. ISSN 0916-8532.
3. CARMIGNIANI, Julie, Borko FURHT, Marco ANISETTI, Paolo CERAVOLO, Ernesto DAMIANI a Misa IVKOVIC. Augmented reality technologies, systems and applications. *Multimedia Tools and Applications* [online]. 2011, **51**(1), 341-377 [cit. 2017-03-25]. DOI: 10.1007/s11042-010-0660-6. ISSN 1380-7501. Dostupné z: <http://link.springer.com/10.1007/s11042-010-0660-6>
4. Sensorama. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-03-23]. Dostupné z: <https://en.wikipedia.org/w/index.php?title=Sensorama&oldid=771722822>
5. SUTHERLAND, Ivan E. *The Ultimate Display* [online]. 1965, , 506-508 [cit. 2017-03-25]. DOI: 10.1.1.97.5951. Dostupné z: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.97.5951&rep=rep1&type=pdf>
6. The Sword of Damocles (virtual reality). In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-03-26]. Dostupné z: [https://en.wikipedia.org/w/index.php?title=The_Sword_of_Damocles_\(virtual_reality\)&oldid=749507756](https://en.wikipedia.org/w/index.php?title=The_Sword_of_Damocles_(virtual_reality)&oldid=749507756)
7. AZUMA, Ronald T. A survey of augmented reality. *Presence: Teleoperators and virtual environments* [online]. MIT Press, 1997, **6**(4), 355-385 [cit. 2017-03-26]. DOI: 10.1162/pres.1997.6.4.355. ISSN 1054-7460. Dostupné z: <http://www.mitpressjournals.org/doi/pdfplus/10.1162/pres.1997.6.4.355>
8. THOMAS, Bruce, Ben CLOSE, John DONOGHUE, John SQUIRES, Phillip De BONDI a Wayne PIEKARSKI. First Person Indoor/Outdoor Augmented Reality Application: ARQuake. *Personal and Ubiquitous Computing* [online]. 2002, **6**(1),

- 75-86 [cit. 2017-09-17]. DOI: 10.1007/s007790200007. ISSN 1617-4909. Dostupné z: <http://link.springer.com/10.1007/s007790200007>
9. VAN KREVELEN, Rick a Ronald POELMAN. A Survey of Augmented Reality Technologies, Applications and Limitations. *The International Journal of Virtual Reality* [online]. 2010, **9**(2), 1-20 [cit. 2017-09-23]. Dostupné z: [https://www.researchgate.net/publication/279867852 A Survey of Augmented Reality Technologies Applications and Limitations](https://www.researchgate.net/publication/279867852_A_Survey_of_Augmented_Reality_Technologies_Applications_and_Limitations)
 10. KOUROUTHANASSIS, Panos E., Costas BOLETIS a George LEKAKOS. Demystifying the design of mobile augmented reality applications. *Multimedia Tools and Applications* [online]. 2015, **74**(3), 1045-1066 [cit. 2017-09-24]. DOI: 10.1007/s11042-013-1710-7. ISSN 1380-7501. Dostupné z: <http://link.springer.com/10.1007/s11042-013-1710-7>
 11. S. K. ONG a A. Y. C. NEE. *Virtual and Augmented Reality Applications in Manufacturing* [online]. London: Springer London, 2004 [cit. 2017-09-23]. ISBN 1447138732.
 12. MEKNI, Mehdi a André LEMIEUX. *Augmented Reality: Applications, Challenges and Future Trends* [online]. 2014, , 205-214 [cit. 2017-09-20]. Dostupné z: <http://www.wseas.us/e-library/conferences/2014/Malaysia/ACACOS/ACACOS-29.pdf>
 13. ZIMMERMAN, Heather T. a Susan M. LAND. Synthesizing perspectives on augmented reality and mobile learning. *TechTrends* [online]. 2014, **58**(1) [cit. 2017-09-28]. Dostupné z: http://www.academia.edu/5890312/Synthesizing_perspectives_on_augmented_reality_and_mobile_learning
 14. Virtual Reality vs. Augmented Reality. In: *Augment News* [online]. 2015 [cit. 2017-09-20]. Dostupné z: <http://www.augment.com/blog/virtual-reality-vs-augmented-reality/>
 15. SMITH, John. Virtual Reality vs. Augmented Reality: A Head-To-Head Comparison. In: *Business 2 Community: Top Trends, News & Expert Analysis* [online]. 2017 [cit. 2017-09-20]. Dostupné z: <http://www.business2community.com/tech-gadgets/virtual-reality-vs-augmented-reality-head-head-comparison-01805086>

16. RASTOGI, Neha. Virtual Reality vs. Augmented Reality: a Comparative Analysis. In: *EngineersGarage: Inspiring Creations* [online]. 2017 [cit. 2017-09-20]. Dostupné z: <https://www.engineersgarage.com/blog/virtual-reality-vs-augmented-reality-comparative-analysis>
17. SIMSARIAN, Kristian T. a Karl-Petter ÅKESSON. *Windows on the World: An example of Augmented Virtuality* [online]. Stockholm, Sweden: Swedish Institute of Computer Science, 1999 [cit. 2017-10-08]. DOI: 10.1.1.97.5951. Dostupné z: <http://citeseerx.ist.psu.edu/viewdoc/versions?doi=10.1.1.27.4281>
18. TAMURA, H., H. YAMAMOTO a A. KATAYAMA. Mixed reality: future dreams seen at the border between real and virtual worlds. *IEEE Computer Graphics and Applications* [online]. 2001, **21**(6), 64-70 [cit. 2017-10-08]. DOI: 10.1109/38.963462. ISSN 02721716. Dostupné z: <http://ieeexplore.ieee.org/document/963462/>
19. Word Lens. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-09-03]. Dostupné z: https://en.wikipedia.org/w/index.php?title=Word_Lens&oldid=793961760
20. Překladač Google. In: *Google Play* [online]. [cit. 2017-09-03]. Dostupné z: <https://play.google.com/store/apps/details?id=com.google.android.apps.translate&hl=cs>
21. Microsoft Translator. In: *Google Play* [online]. [cit. 2017-09-03]. Dostupné z: <https://play.google.com/store/apps/details?id=com.microsoft.translator&hl=cs>
22. SOOD, Raghav. Applications of Augmented Reality. *Pro Android Augmented Reality* [online]. Berkeley, CA: Apress, 2012, , 1 [cit. 2017-09-20]. DOI: 10.1007/978-1-4302-3946-8_1. ISBN 978-1-4302-3945-1. Dostupné z: http://link.springer.com/10.1007/978-1-4302-3946-8_1
23. VLAHAKIS, V., M. IOANNIDIS, J. KARIGIANNIS, et al. Archeoguide: an augmented reality guide for archaeological sites. *IEEE Computer Graphics and Applications* [online]. 2002, **22**(5), 52-60 [cit. 2017-09-20]. DOI: 10.1109/MCG.2002.1028726. ISSN 0272-1716. Dostupné z: <http://ieeexplore.ieee.org/document/1028726/>

24. SHIN, Jennifer. Governments Investing in AR & VR for Education. In: *Augment News* [online]. 2017 [cit. 2017-09-20]. Dostupné z: <http://www.augment.com/blog/govt-investing-arvr-education/>
25. CAVALLARO, R. The FoxTrax hockey puck tracking system. *IEEE Computer Graphics and Applications* [online]. **17**(2), 6-12 [cit. 2017-09-02]. DOI: 10.1109/38.574652. ISSN 02721716. Dostupné z: <http://ieeexplore.ieee.org/document/574652/>
26. STRAUSS, Paul. Mini Augmented Reality Ads Hit Newstands. In: *Technabob* [online]. 2008 [cit. 2017-09-03]. Dostupné z: <https://technabob.com/blog/2008/12/17/mini-augmented-reality-ads-hit-newstands/>
27. ARQuake: Interactive Outdoor Augmented Reality Collaboration System. In: *Wearable Computer Lab: University of South Australia* [online]. [cit. 2017-09-17]. Dostupné z: <http://wearables.unisa.edu.au/projects/arquake/>
28. SWATMAN, Rachel. Pokémon Go catches five new world records. In: *Guinness World Records* [online]. 2016 [cit. 2017-08-20]. Dostupné z: <http://www.guinnessworldrecords.com/news/2016/8/pokemon-go-catches-five-world-records-439327>
29. CARTER, Marcus a Eduardo VELLOSO. Some places should be off limits for games such as Pokémon GO. In: *The Conversation* [online]. 2016 [cit. 2017-08-26]. Dostupné z: <https://theconversation.com/some-places-should-be-off-limits-for-games-such-as-pokemon-go-62341>
30. D'ANASTASIO, Cecilia. You Can No Longer Catch Pokémon At Hiroshima's Memorial Or The Holocaust Museum. In: *Kotaku* [online]. 2016 [cit. 2017-08-26]. Dostupné z: <http://kotaku.com/you-can-no-longer-catch-pokemon-at-hiroshima-1784985508>
31. AKHTAR, Allana. Holocaust Museum, Auschwitz want Pokémon Go hunts out. In: *USA Today* [online]. 2016 [cit. 2017-08-26]. Dostupné z: <https://www.usatoday.com/story/tech/news/2016/07/12/holocaust-museum-auschwitz-want-pokmon-go-hunts-stop-pokmon/86991810/#>
32. Ingress (video game). In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-08-27]. Dostupné z:

- [https://en.wikipedia.org/w/index.php?title=Ingress \(video game\)&oldid=795149253](https://en.wikipedia.org/w/index.php?title=Ingress_(video_game)&oldid=795149253)
33. *ADC 2 Overall Winners* [online]. 2010 [cit. 2017-08-27]. Dostupné z: https://web.archive.org/web/20100623234336/http://code.google.com/android/adc/gallery_winners.html
34. SpecTrek. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-08-27]. Dostupné z: <https://en.wikipedia.org/w/index.php?title=SpecTrek&oldid=755680286>
35. *Lyteshot: Live Action Gaming* [online]. 2015 [cit. 2017-09-02]. Dostupné z: <http://www.lyteshot.com/>
36. BECKMAN, Mariah. LyteShot Interview with CEO Mark Ladd & CTO Tom Ketola. In: *Gizmos, Games, News & Reviews / GIZORAMA* [online]. 2014 [cit. 2017-09-02]. Dostupné z: <https://www.gizorama.com/2014/interview/lyteshot-interview-with-ceo-lark-ladd-cto-tom-ketola>
37. Makers and Game Designers - Lyteshot. *Lyteshot* [online]. [cit. 2017-09-02]. Dostupné z: <http://www.lyteshot.com/makers/>
38. KYOTO, Kate. Microsoft HoloLens Specifications. *SizeScreens.com: All Smart Gadgets and VR Headsets are Here* [online]. 14. 7. 2016 [cit. 2018-01-20]. Dostupné z: <https://www.sizescreens.com/microsoft-hololens-specifications/>
39. KYOTO, Kate. Lenovo Glass C220 Specifications. *SizeScreens.com: All Smart Gadgets and VR Headsets are Here* [online]. 10. 1. 2018 [cit. 2018-01-20]. Dostupné z: <https://www.sizescreens.com/lenovo-glass-c220-specifications/>
40. GLASS: *Google Glass Help* [online]. [cit. 2018-01-18]. Dostupné z: <https://support.google.com/glass/>
41. *Microsoft HoloLens: The leader in mixed reality technology* [online]. [cit. 2018-01-18]. Dostupné z: <https://www.microsoft.com/en-us/hololens>
42. FROM POCKET TO PC TO HOME: LENOVO'S CES 2018 LINEUP MAKES REALITY BETTER. In: *Lenovo Newsroom* [online]. 9. 1. 2018 [cit. 2018-01-18]. Dostupné z: <http://news.lenovo.com/news-releases/from-pocket-to-pc-to-home-lenovos-ces-2018-lineup-makes-reality-better.htm>
43. DORMEHL, Luke. The \$30 ZapBox has the chance to do for mixed reality what Google Cardboard did for VR. In: *Digital Trends: Technology News and Product*

- Reviews* [online]. 17. 11. 2016 [cit. 2018-01-20]. Dostupné z: <https://www.digitaltrends.com/virtual-reality/zapbox-kickstarter-campaign/>
44. KYOTO, Kate. *ZapBox. SizeScreens.com: All Smart Gadgets and VR Headsets are Here* [online]. 14. 12. 2016 [cit. 2018-01-20]. Dostupné z: <https://www.sizescreens.com/zapbox/>
45. TEMPLETON, Graham. *What's Up with Augmented Reality Contact Lenses?* In: *Inverse* [online]. 22. 5. 2017 [cit. 2018-01-20]. Dostupné z: <https://www.inverse.com/article/31034-augmented-reality-contact-lenses>
46. Operating system market share: Mobile/Tablet Operating System Market Share. *NetMarketShare* [online]. 2017 [cit. 2017-12-26]. Dostupné z: [https://www.netmarketshare.com/operating-system-market-share.aspx?options={"filter"%3A{"%24and"%3A\[{"deviceType"%3A{"%24in"%3A\["Mobile"%2C"Tablet"\]}\]}}%2C"dateLabel"%3A"Custom"%2C"attributes"%3A"share"%2C"group"%3A"platform"%2C"sort"%3A{"share"%3A-1}%2C"id"%3A"platformsDesktop"%2C"dateInterval"%3A"Monthly"%2C"dateStart"%3A"2017-11"%2C"dateEnd"%3A"2017-11"%2C"segments"%3A"-1000"}](https://www.netmarketshare.com/operating-system-market-share.aspx?options={)
47. JACKSON, Dean. *Next-generation 3D Graphics on the Web.* In: *WebKit* [online]. 2017, 7. 2. 2017 [cit. 2017-12-30]. Dostupné z: <https://webkit.org/blog/7380/next-generation-3d-graphics-on-the-web/>
48. WebGL - 3D Canvas graphics. *Can I use.: Support tables for HTML5, CSS3, etc* [online]. [cit. 2018-01-02]. Dostupné z: <https://caniuse.com/#feat=webgl>
49. WebGL 2.0. *Can I use.: Support tables for HTML5, CSS3, etc* [online]. [cit. 2017-01-02]. Dostupné z: <https://caniuse.com/#feat=webgl2>
50. JOHANSSON, Tim. *Taking the canvas to another dimension.* In: *Tim's blog* [online]. 13. 11. 2007 [cit. 2017-12-30]. Dostupné z: <https://web.archive.org/web/20140226155143/http://my.opera.com/timjoh/blog/2007/11/13/taking-the-canvas-to-another-dimension>
51. GILBERT, Jeff a Belén ALBEZA. *WebGL 2 lands in Firefox.* In: *Mozilla Hacks: the Web developer blog* [online]. 24. 1. 2017 [cit. 2017-12-30]. Dostupné z: <https://hacks.mozilla.org/2017/01/webgl-2-lands-in-firefox/>

52. WebGLTransformFeedback. *MDN Web Docs: Resources for developers, by developers*. [online]. [cit. 2017-12-31]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/API/WebGLTransformFeedback>
53. JACKSON, Dean a Jeff GILBERT, ed. WebGL 2 Specification. In: *The Khronos Group Inc: WebGL Overview* [online]. 11. 4. 2017 [cit. 2017-12-30]. Dostupné z: <https://www.khronos.org/registry/webgl/specs/2.0/>
54. WebGPU demos. *WebKit* [online]. [cit. 2017-12-31]. Dostupné z: <https://webkit.org/demos/webgpu/>
55. DeviceOrientation & DeviceMotion events. *Can I use...: Support tables for HTML5, CSS3, etc* [online]. [cit. 2018-01-02]. Dostupné z: <https://caniuse.com/#feat=deviceorientation>
56. Detecting device orientation. *MDN Web Docs: Resources for developers, by developers*. [online]. 2017 [cit. 2017-12-29]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/API/Detecting_device_orientation
57. DeviceMotionEvent.accelerationIncludingGravity. *MDN Web Docs: Resources for developers, by developers*. [online]. [cit. 2017-12-29]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/API/DeviceMotionEvent/accelerationIncludingGravity>
58. Prefer Secure Origins For Powerful New Features. *The Chromium Projects* [online]. [cit. 2017-12-27]. Dostupné z: <https://www.chromium.org/Home/chromium-security/prefer-secure-origins-for-powerful-new-features>
59. GetUserMedia/Stream API. *Can I use...: Support tables for HTML5, CSS3, etc* [online]. 2018 [cit. 2017-01-02]. Dostupné z: <https://caniuse.com/#feat=stream>
60. WebVR API. *MDN Web Docs: Resources for developers, by developers*. [online]. [cit. 2017-12-31]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/API/WebVR_API
61. *WebVR: Bringing Virtual Reality to the Web* [online]. [cit. 2017-12-31]. Dostupné z: <https://webvr.info/>

62. WebVR API. *Can I use.: Support tables for HTML5, CSS3, etc* [online]. [cit. 2018-01-02]. Dostupné z: <https://caniuse.com/#feat=webvr>
63. *WebVR Rocks: Your guide to Virtual Reality in the browser* [online]. [cit. 2018-01-01]. Dostupné z: <https://webvr.rocks/>
64. Navigator.getVRDisplays(). *MDN Web Docs: Resources for developers, by developers* [online]. [cit. 2018-01-01]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/API/Navigator/getVRDisplays>
65. VRIGNAUD, Andre a Lars BERGSTROM. Bringing Mixed Reality to the Web. In: *The Mozilla Blog: Dispatches from the Internet frontier*. [online]. 20. 10. 2017 [cit. 2018-01-01]. Dostupné z: <https://blog.mozilla.org/blog/2017/10/20/bringing-mixed-reality-web/>
66. WebXR Device API Explained. *Webxr: Repository for the WebXR Device API Specification*. [online]. [cit. 2018-01-02]. Dostupné z: <https://github.com/immersive-web/webxr/blob/master/explainer.md>
67. Web Sockets. *Can I use.: Support tables for HTML5, CSS3, etc* [online]. [cit. 2018-01-12]. Dostupné z: <https://caniuse.com/#feat=websockets>
68. KOŠŤÁK, Milan. *Zobrazení grafické prostorové scény v prostředí WWW: Technologie WebGL*. Hradec Králové, 2014. Bakalářská práce. Univerzita Hradec Králové. Vedoucí práce Ing. Bruno Ježek, Ph.D.

9. Příloha

Přiložené DVD má následující strukturu a obsah:

- Adresář „*implementace*“ obsahuje soubory s implementací praktické části práce
- Soubor „*implementace/index.html*“ je hlavním rozcestníkem pro jednotlivé navržené testovací úlohy
- Adresář „*text*“ obsahuje tento text ve formátech „.docx“ a „.pdf“
- Soubor „*video.mp4*“ obsahuje 50 sekundové video s ukázkou všech 3 navržených testovacích aplikací