

MASARYKOVÁ  
UNIVERZITA

PŘÍRODOVĚDECKÁ FAKULTA

**Webová virtuální realita: nový způsob prezentace  
geoprostorových dat**

Diplomová práce

**Jan Horák**

Vedoucí práce: RNDr. Lukáš Herman, Ph.D.

Geografický ústav

**BRNO 2024**

## BIBLIOGRAFICKÝ ZÁZNAM

<b>Autor/Autorka:</b>	Bc. Jan Horák
	Přírodovědecká fakulta, Masarykova univerzita
	Geografický ústav
<b>Název práce:</b>	Webová virtuální realita: nový způsob prezentace geoprostorových dat
<b>Studijní program:</b>	Geografická kartografie a geoinformatika
<b>Studijní obor:</b>	Geografická kartografie a geoinformatika
<b>Vedoucí práce:</b>	RNDr. Lukáš Herman, Ph.D.
<b>Akademický rok:</b>	2022/2023
<b>Počet stran:</b>	xx
<b>Klíčová slova:</b>	GIS, VR, XR, WEB, Virtuální realita, Uživatelské testování, 3D

## BIBLIOGRAFICKÝ ENTRY

<b>Author:</b>	Bc. Jan Horák Faculty of Science, Masaryk University Department of Geography
<b>Title of Thesis:</b>	Web virtual reality: a new way of presenting geospatial data
<b>Degree Programme:</b>	Cartography and geoinformatics
<b>Field of Study:</b>	Cartography and geoinformatics
<b>Supervisor:</b>	RNDr. Lukáš Herman, Ph.D.
<b>Academic Year:</b>	2022/2023
<b>Number of Pages:</b>	xx
<b>Keywords:</b>	GIS, VR, XR, WEB, Virtual reality, User testing, 3D

## **ABSTRAKT**

Tato práce se zabývá..

## **ABSTRACT**

Diploma thesis considers...

## ZADÁNÍ DIPLOMOVÉ PRÁCE

**Student:** Bc. Jan Horák

**Studijní program:** Geografická kartografie a geoinformatika

**Studijní obor:** Geografická kartografie a geoinformatika

Ředitel Geografického ústavu Přírodovědecké fakulty MU Vám ve smyslu Studijního a zkušebního řádu MU určuje bakalářskou práci s tématem:

**Webová virtuální realita: nový způsob prezentace geoprostorových dat**

**Web virtual reality: a new way of presenting geospatial data**

Zásady pro vypracování:

Práce bude zaměřena na technologie pro tvorbu virtuální reality v rámci webového prostředí (např. WebXR, Three.js). Tyto technologie jsou podporovány na různých hardwarových a softwarových platformách, jejich funkcionalita se může v různých podmínkách lišit. Na základě srovnání dostupných technologických bude vybrána technologie pro vytvoření vlastní vizualizace prostorových dat z vybrané aplikativní oblasti. Vytvořená vizualizace bude následně uživatelsky evaluována a zhodnocena tak její funkcionalita.

Pro naplnění hlavního cíle diplomové práce postupujte přes následující dílčí cíle:

1. Popis a analýza technologií pro tvorbu virtuální reality v rámci webového prostředí
2. Praktické porovnání konkrétních technologií na různých hardwarových a softwarových platformách
3. Návrh a implementace vlastní aplikace na principech virtuální reality
4. Uživatelské ověření vytvořené aplikace
5. Diskuse zjištěných výsledků a závěr

Rozsah grafických prací: podle potřeby

Rozsah průvodní zprávy: cca 60 až 80 stran

Seznam odborné literatury:

BUTCHER, P. W. S., JOHN, N. W., RITSOS, P. D. (2021): VRIA: A Web-Based Framework for Creating Immersive Analytics Experiences. IEEE Transactions on Visualization and Computer Graphics, roč. 27, č. 7, s. 3213–3225. <http://doi.org/10.1109/TVCG.2020.2965109>

LAKONSO, D., ADITYA, T. (2019): Utilizing A Game Engine for Interactive 3D Topographic Data Visualization. ISPRS International Journal of Geo-Information, roč. 8, č. 8. <https://doi.org/10.3390/ijgi8080361>

RZESZEWSKI, M., ORYLSKI, M. (2021): Usability of WebXR Visualizations in Urban Planning. ISPRS International Journal of Geo-Information, roč. 10, č. 11. <https://doi.org/10.3390/ijgi10110721>

STACHOŇ, Z., KUBÍČEK, P. HERMAN, L. (2020): Virtual and Immersive Environments. Wilson, J. P.: The Geographic Information Science & Technology Body of Knowledge. Ithaca, New York, UCGIS. <https://gistbok.ucgis.org/bok-topics/virtual-and-immersive-environments>

SEO, D., YOO, B. (2020): Interoperable information model for geovisualization and interaction in XR environments, International Journal of Geographical Information Science, roč. 34, č. 1. s. 1–30. <https://doi.org/10.1080/13658816.2019.1706739>

ŠTĚRBA, Z., ŠAŠINKA, Č., STACHOŇ, Z., ŠTAMPACH, R., MORONG, K. (2015): Selected Issues of Experimental Testing in Cartography. Masaryk University, Brno, 107 s., ISBN 978-80-210-7909-0.

*Jazyk závěrečné práce:* čeština

*Vedoucí bakalářské práce:* RNDr. Lukáš Herman, Ph.D.

*Datum zadání diplomové práce:* 25. 1. 2022

*Datum odevzdání diplomové práce:* dle harmonogramu

RNDr. Vladimír Herber, CSc.  
pedagogický zástupce ředitele ústavu

## **PODĚKOVÁNÍ**

Na tomto místě bych chtěl poděkovat vedoucímu diplomové práce RNDr. Lukáši Hermanovi, Ph.D za vedení práce a předevím za trpělivost a ochotu nasměrovat. Velké díky patří kolegům z Kanceláře architekta města, s jejichž pomocí byly získány potřebné znalosti. Dále bych chtěl poděkovat členům vývojového týmu aplikace Wonderland engine, četné a podrobné rady. V neposlední řadě patří obrovské díky mé rodině a blízkým za neustálou podporu.

## **PROHLÁŠENÍ**

Prohlašuji, že jsem svoji bakalářskou/diplomovou práci vypracoval samostatně pod vedením RNDr. Lukáše Hermana, Ph.D. a s využitím informačních zdrojů, které jsou v práci citovány.

Brno 31. 12. 2023

---

Jméno Příjmení

# OBSAH

<b>1</b>	<b>ÚVOD .....</b>	<b>10</b>
1.1	CÍLE PRÁCE.....	10
1.2	METODY VYPRACOVÁNÍ .....	11
<b>2</b>	<b>REŠERŠE .....</b>	<b>12</b>
<b>3</b>	<b>TEORETICKÉ ZÁKLADY .....</b>	<b>14</b>
3.1	VIRTUÁLNÍ REALITA.....	14
3.1.1	<i>Komunikační medium.....</i>	15
3.1.2	<i>VR a kartografie .....</i>	16
3.2	SYSTÉM VIRTUÁLNÍ REALITY .....	18
3.2.1	<i>Hardware .....</i>	19
3.2.2	<i>Software .....</i>	19
3.3	KOMPONENTY SYSTÉMU VIRTUÁLNÍ REALITY.....	20
3.3.1	<i>Výstup – Output .....</i>	20
3.3.2	<i>Vstup – Input .....</i>	21
3.4	PERCEPCE .....	25
3.4.1	<i>Percepce vzdálenosti.....</i>	25
3.4.2	<i>Percepce pohybu .....</i>	26
3.5	3D MODELOVÁNÍ KONCEPTY A PRINCIPY.....	27
3.5.1	<i>LOD.....</i>	27
3.5.2	<i>Problém měřítka.....</i>	28
3.5.3	<i>Datové modely a formáty .....</i>	30
3.6	3D GRAFIKA .....	33
3.6.1	<i>Rendering pipeline .....</i>	33
3.6.2	<i>Komponenty 3D vizualizace .....</i>	34
3.6.3	<i>Vývoj.....</i>	37
3.6.4	<i>Výkon .....</i>	37
<b>4</b>	<b>ANALÝZA TECHNOLOGIÍ .....</b>	<b>38</b>
4.1	TAXONOMIE TECHNOLOGIÍ .....	38
4.2	WEBOVÝ VÝVOJ .....	39
4.2.1	<i>Web API.....</i>	40
4.2.2	<i>Prohlížeče .....</i>	42
4.2.3	<i>Vývojářské nástroje .....</i>	42
4.3	ANALÝZA TECHNOLOGIÍ.....	43
4.3.1	<i>Geoprostorová řešení.....</i>	44
4.3.2	<i>Vykreslovací enginy.....</i>	47
4.3.3	<i>VR frameworky.....</i>	49
4.3.4	<i>Herní enginy .....</i>	50
4.3.5	<i>Služby .....</i>	52
4.3.6	<i>Optimalizační nástroje .....</i>	52
4.4	VÝBĚR TECHNOLOGIE .....	53
<b>5</b>	<b>NÁVRH A IMPLEMENTACE VLASTNÍ APLIKACE .....</b>	<b>56</b>
5.1	NÁVRH.....	56
5.2	DATA .....	57
5.2.1	<i>Specifika vstupních dat.....</i>	58
5.3	VÝVOJ APLIKACE .....	62

5.3.1	<i>Transformace dat</i> .....	62
5.3.2	<i>Tvorba VP</i> .....	67
5.3.3	<i>CI / CD</i> .....	73
<b>6</b>	<b>UŽIVATELSKÉ TESTOVÁNÍ</b> .....	<b>75</b>
6.1	NÁVRH TESTOVÁNÍ .....	75
6.1.1	<i>Průběh testování</i> .....	75
6.1.2	<i>Průchod VP</i> .....	76
6.2	VÝSLEDKY .....	77
6.2.1	<i>Identifikace</i> .....	77
6.2.2	<i>Úkoly</i> .....	78
6.2.3	<i>Ovládání</i> .....	81
6.2.4	<i>Osobní názor</i> .....	82
<b>7</b>	<b>DISKUZE</b> .....	<b>83</b>
<b>8</b>	<b>ZÁVĚR</b> .....	<b>85</b>

# 1 ÚVOD

Tvorba virtuálních realit (alternativních světů) má kořeny hluboko v lidské historii. Při dostatečné abstrakci je možné vnímat virtuální realitu jako médium komunikace a způsob zobrazení reálných či fiktivních světů. S touto definicí je možné nahlížet na historii virtuální reality skrze prizma obrazů na stěnách jeskyň, vyprávění reálných i imaginárních událostí, malby na plátnech, fotografie, filmy, videohry – a dospět až k současné virtuální realitě.

Tato myšlenka vede k názoru, že vývoj technologie je pevně provázán s rozvojem médií určených pro tvorbu různých forem virtuální reality. Dnes umožňuje práci s tvorbou virtuální reality především výpočetní technika, která slouží jako médium i nástroj pro lidskou komunikaci ve virtuálním prostoru. Za účelem tvorby kvalitní virtuální reality je nutná znalost nástrojů pro její tvorbu. Stejně jako malíř musel znát výhody a nevýhody druhů plátna, barev a štětců musí dnes člověk který chce tvořit virtuální realitu znát jemu dostupné technologie a zvolit nejlepší pro jeho účel.

Virtuální realita nemusí znamenat čistě fiktivní světy, ale v případě prostorové informace je primárním zájmem imitace reálného světa ve světe digitálním. Na druhou stranu virtuální realita se nemusí snažit o kopii reálného světa, ale může cílit k tvorbě zjednodušené verze reality, kdy je kladen důraz na specifickou charakteristiku, kterou v realitě není schopné pozorovat (např. zobrazení statistických dat pomocí prostorových proměnných).

V případě pohledu na virtuální realitu jako způsobu komunikace se nabízí myšlenka jejího propojení s internetovým prostředím, které je možné považovat za stále rostoucí způsob lidské komunikace a interakce. Toto propojení přináší jak dostupnost, tak i možnost sdílení, interakce, popř. vzájemné kolaborace což odborná literatura považuje za stěžejní pilíře virtuální reality.

Práce nejdříve představuje obecné teoretické základy virtuální reality. Následně klíčovou částí je analýza soudobých webových technologií a jejich zhodnocení spolu s pilotní implementací webové aplikace zobrazující geografická data. Finální částí je následně uživatelské zhodnocení výsledné pilotní aplikace.

## 1.1 Cíle práce

### Hlavní cíle práce

- Analýza a zhodnocení webových technologií umožňující publikaci prostorových dat v rámci virtuální reality.
- Vývoj pilotní aplikace pomocí vybraných technologií.

Hlavním cílem práce je analýza a následné zhodnocení soudobých technologií, a to jak hardwarových (HMD, mobilní telefony, stolní počítače) tak softwarových (webové prohlížeče, knihovny, frameworky). Dále také jejich propojení, umožňující tvorbu VP (virtuálních prostředí).

### Vedlejší cíle práce

- Uživatelské zhodnocení pilotní aplikace
- Odborný přehled metod tvorby VP ve webovém prostředí.

Vedlejším cílem práce je vytvoření a následná uživatelská evaluace funkcionality webové aplikace využívající prostředků virtuální reality, jakožto nástroje k prezentaci geografických dat.

## **1.2 Metody vypracování**

První část práce se zaměřila na zpracování teoretické problematiky virtuální reality. Využito bylo primárně odborné literatury napříč vědními obory. V případě teorie počítačové grafiky a 3D modelování byla mimo odbornou literaturu v řadě případů využita dokumentace analyzovaných softwarových řešení.

# todo - dopsat přehled metodiky

## 2 REŠERŠE

Kapitola představující základní výchozí literaturu pro vypracování tématu. Nejedná se o kompletní seznam použitých zdrojů, především elektronické zdroje jsou citovány v textu relevantních kapitol.

Za účelem získání obecného přehledu o problematice je vhodný průzkum obecných publikací z oblastí počítačové grafiky (Žára, Beneš, Felkel 2005; Marschner et al. 2021), geoinformační vědy (Guo, Goodchild, Annoni 2020; Bolstad 2019; Kresse, Danko 2012; Longley et al. 2015), kartografické geo-vizualizace (Slocum 2014; Çöltekin et al. 2020a; Christophe 2020; Dykes, MacEachren, Kraak 2005), webových technologií (Dorman 2020) a virtuální / rozšířené reality (Milgram, Kishino 1994; LaValle 2020; Sherman, Craig 2019; Mazuryk, Gervautz 1999).

Koncept virtuální reality obecně představují výše uvedené publikace. Z hlediska geoinformatiky a geografie je vhodnější koncept virtuálních geografických prostředí, tedy VGE (*virtual geographic environment*) tento způsob pohledu představují (Stachon, Kubicek, Herman 2020; Çöltekin et al. 2020b; Batty 1997; Lin, Batty 2011; MacEachren et al. 1999; Blokdyk 2018; Lin et al. 2013).

Prerekvizitou úspěšné geoprostorové vizualizace, jak tradiční, tak ve virtuálních prostředí, je podrobné porozumění vstupním datům tedy datovým modelům, metodám zpracování a výměny mezi technologiemi (Keil et al. 2021). Základy modelování dat řeší (Abdul-Rahman, Pilouk 2008). Problematiku převodu 2D do 3D dat rozebírá (Halik 2018). Kompletní postup od získání geoprostorových dat přes integraci po jejich vizualizaci představují (Zhao et al. 2019; Laksono, Aditya 2019; Herman 2011; 2014; Buyuksalih et al. 2017; Keil et al. 2021). Zaměření na technické aspekty sběru a následně publikace 3D pomocí webového informačního systému řeší (Cibula 2021). Mimo samotná data je také nutné mít na paměti kartografická pravidla a principy při vizualizaci především 3D dat (Pegg 2008).

Proto aby vizualizace plnila svůj účel je nutné, aby byla přístupná uživatelům. Přístupnost v kontextu této práce představuje publikace vizualizace ve webovém prostředí. Problematikou rozšířené reality ve webovém prostředí se zabývá (MacIntyre, Smith 2018). Podrobný návod tvorby VR prostředí na webu představuje (Baruah 2021). (Butcher, John, Ritsos 2021) představuje webový framework pro tvorbu obecných vizualizací dat ve VR na webu. V případě VGE je často řešena problematika distribuovaných *kolaborativních* prostředí v rámci rozšířené reality. Obecně tuto problematiku řeší (Lee, Yoo 2021; Šašinka et al. 2019), v geoinformačním kontextu pak (Sermet, Demir 2021).

Za účelem vhodného výběru technologie pro vývoj XR aplikace pro prezentaci geoprostorových dat je vhodný široký průzkum případových studií a jejich následná typologie na základě tematického zaměření, ale i využité technologie. Vizualizaci terénu pomocí herních enginů řeší (Mat et al. 2014). Komplexnější scény za pomocí herních enginů pak představují (Ugwitz, Stachoň, Kubicek 2021; Laksono, Aditya 2019; Keil et al. 2021). Vizualizace terénu je řešena pomocí webových technologií v (Herman, Řezník 2015).

Široké využití nachází 3D vizualizace a XR v případě urbánních prostředí a to v různých odvětvích např. urbánní plánování, architektura, meteorologie aj. Vizualizaci 3D city modelů napříč projekty shrnuje (Julin et al. 2018; Herman 2014). Koncept AR aplikace pro terénní urbánní plánování shrnuje (Cirulis, Brigmanis 2013). V oblasti meteorologie využívá 3D vizualizace (Gautier, Christophe, Brédif 2020; Gautier, Brédif, Christophe 2020), v kontextu plánování umístění větrných elektráren pak (Rafiee et al. 2018). V případě územního plánování se využitím 3D vizualizace zabývá (Judge, Harrie 2020). 3D webové vizualizace seismických

hazardů infrastruktury využívají (Mazzei, Quaroni 2022). V rámci územního plánování je VR 3D vizualizace často zmiňována v kontextu zvýšení participace veřejnosti na vývoji územního plánu. Autoři (Judge, Harrie 2020; Onyimbi, Koeva, Flacke 2018; Rzeszewski, Orylski 2021) považují 3D vizualizace za přínosné v tomto ohledu. Virtuální prostředí v kontextu geografie jsou často zmiňována i v oblasti výuky. Vývoj systému pro sběr a vizualizaci dat v rámci virtuální reality představuje (Bernardes et al. 2018). Vývoj aplikací pro web zahrnující propojení s geoprostorovou informací se zabývá (Peňák 2017).

Součástí vývoje aplikace pro vizualizaci geo-dat je uživatelské zhodnocení. Metodiku a aspekty hodnocení map rozebírají (Sterba et al. 2015). Metodiku pro obecné uživatelské testování použitelnosti systémů pak představuje (Brooke 1995).

### 3 TEORETICKÉ ZÁKLADY

V teoretické části je nutné objasnit podstatné pojmy a koncepty. Teoretická část se zabývá především tématikou virtuální reality, a to z pohledu koncepčního (obecného, komunikačního), inženýrského, popř. technického, ale i fyziologického (percepčního). Následně je prozkoumána oblast 3D modelování.

#### 3.1 Virtuální realita

Definování konceptu virtuální reality je obtížné, především kvůli jeho širokému záběru a rychle se měnícím technologickým trendům. Definice je navíc rozdílně interpretována napříč obory, které se VR zabývají. Sherman a Craig definují virtuální realitu následovně:

*Virtual reality: a medium composed of interactive computer simulations that sense the participant's position and actions and replace or augment the feedback to one or more senses, giving the feeling of being mentally immersed or present in the simulation (a virtual world). (Sherman, Craig 2019)*

Důležitým konceptem je pak také chápání virtuální reality jako zážitku (*experience*), jelikož virtuální realita je reálná až v případě, kdy jí někdo (Účastník) zažívá / vnímá. Jakožto zážitek definuje virtuální realitu i LaValle následovně:

*Inducing targeted behavior in an organism by using artificial sensory stimulation while the organism has little or no awareness of the interference. (LaValle 2020)*

MacEachren definuje virtuální realitu pomocí 4 I virtuální reality (MacEachren et al. 1999) a to **Imerze, Interaktivita, Informační intenzita a Inteligence objektů**.

Definice Shermana a Craiga pohlížející na virtuální realitu více z pohledu zážitku stejně jako LaValle, tato definice je založena na 5 klíčových elementech:

1. Účastník
2. Tvůrce
3. Virtuální svět / prostředí
4. Imerze
5. Interaktivita

**Virtuálním světem** Sherman a Craig chápou kolekci objektů, pravidel a vztahů mezi objekty v prostoru. Jedná se tedy o jakýkoliv virtuální svět, který je prezentován **účastníkovi** skrze médium (způsob komunikace mezi účastníky) virtuální reality. V případě, že virtuální svět je vytvořen na základě reálného / geografického, je možné mluvit o virtuálním geografickém prostředí (VGE). **Imerzi** Sherman a Craig rozdělují na **mentální** (stav zapojení do virtuální reality) a **fyzickou** (stimulace senzorů). Dohromady imerzi pak definují jakožto: pocit bytí ve virtuálním prostředí. LaValle definuje stejný koncept pomocí termínu *Awareness*: tedy míru do jaké je uživatel oklamán, že je součástí virtuální reality. **Interaktivita** je podle Shermana a Craiga klíčovým prvkem k autenticitě virtuální reality. LaValle interakci konkretizuje pomocí toho, zdali **účastník** má vliv na virtuální realitu, a to na *open-loop* (nemá vliv) a *closed-loop* (má vliv). Míra interakce se liší podle zvoleného média (technologie), pomocí kterého je virtuální realita vytvořena. MacEachren odlišuje koncepty **informační intenzity** a **inteligence objektů**, a to především proto, že se tyto koncepty vztahují více k VGE nežli

k obecnému konceptu virtuální reality. **Informační intenzitou** je myšlena míra abstrakce vytvořeného virtuálního prostředí od prostředí reálného (viz. kap. **Error! Reference source not found.**). **Inteligencí objektů** se pak chápe míra do jaké jsou objekty ve virtuálním prostředí schopny interakce na základě různých vjemů (např. vzdálenost od účastníka aj.) (Stachon, Kubicek, Herman 2020).

Důležité je objasnit mimo definice i užívanou terminologii. LaValle pod termín virtuální reality řadí termíny AR (*Augmented reality*), MR (*Mixed reality*) a XR (*Extended reality*). Tyto termíny jsou obecně vnímány jako distinktivní body na *virtuálním kontinuu* (Milgram, Kishino 1994). MR a XR jsou pak považovány za generalizace VR, AR a jiných termínů nacházejících se na virtuálním kontinu. XR je považován za novější termín, který zahrnuje zbylé termíny více z pohledu technologického než koncepčního. Na konkrétní definice se názory napříč literaturou různí (Çöltekin et al. 2020b).

Někteří autoři rozlišují mezi termíny virtuální realita a virtuální prostředí. Termín virtuální realita se přesněji nevztahuje k napodobení reality (ale k zážitku, který se pro uživatele jeví jako reálný). Přestože slovo "realita" může vyvolat dojem napodobení skutečnosti, objevil se termín virtuální prostředí. Termín vznikl, protože s využitím vizualizačního prostředí lze také zobrazit fiktivní (nebo plánovaná) prostředí, a tak termín "prostředí" lépe zahrnuje rozsah činností, které lze v takovém vizualizačním prostředí provádět (Çöltekin et al. 2020a).

Dalším z pohledů na definování VR je dělení na základě míry imerze na kategorie (Riva 2006; Kvarda 2020; Burdea, Coiffet 2003):

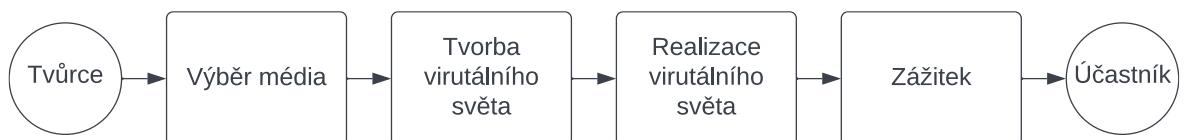
1. Desktop VR
2. Teleprezence
3. CAVE
4. Imerzivní VR (VR)
5. Rozšířená realita (AR)
6. Smíšená realita (MR)

První dvě kategorie nejsou považovány za imerzní. Technologie CAVE umožňující projekci virtuálního světa na přímé fyzické okolí již za imerzní považovaná je. V případě VR, AR a MR se jedná o kombinace práce s reálným a virtuálním světem. (Kvarda 2020; Burdea, Coiffet 2003) Záměrem této práce je primárně imerzní VR tedy systém zobrazující virtuální svět ve virtuálních brýlích (HMD). V imerzním VR je míra imerze určována především úrovní, do jaké umožňuje hardware a software naplnit tvorbu virtuálního prostředí, interaktivitu informační intenzity atd. Míra imerze tedy závisí na tom, jaké smysly a do jaké míry jsou ovlivněny virtuální realitou. Míra imerze je dále ovlivněna kompetencí hardwarového zařízení (viz. kap. 3.3.1).

### 3.1.1 Komunikační medium

Virtuální realita je médium, tudíž je možné ji chápout jako formu mezilidské komunikace a dále zkoumat její vztah k dalším formám komunikace (Sherman, Craig 2019). V oboru kartografie je možné obdobné definice najít ve dvou teoretických koncepcích kartografie, a to v informační a komunikační teorii mapy, které byly rozpracovány již 60. letech minulého století (Kubíček, Stachoň 2009). Teorie kartografické komunikace představuje mapu jako prostředek komunikace, kde tvůrce je „odesílatelem“ informace a čtenář příjemce (Kubíček, Stachoň 2009). Sherman a Craig tuto myšlenku představují obecněji na různých formách medií. Společný je však fakt, že se jedná o lineární proces (Koláčný 1969). Tvůrce myšlenky si vybere

médium (film, mapa, virtuální realita atd. ), následně vytvoří virtuální svět (příběh, reprezentace prostoru na mapě, reprezentace prostoru v počítači), který poté realizuje pomocí technologií příslušící danému médiu, čímž vytváří pro účastníka jedinečný zážitek (Sherman, Craig 2019).



Obr. 1 Základní model komunikace skrze médium. (upraveno podle: Sherman, Craig 2019)

Výše zmíněný model, však není možné zcela aplikovat v případě přidání konceptu interaktivnosti a vlivu účastníka na tvorbu zážitku. Stejná problematika se vyskytuje i v dalších názorech na teorii kartografické komunikace (Slocum 2014; MacEachren 2004).

Kubíček a Stachoň představují myšlenku rozšíření komunikačních možností mapy skrze nové technologie (web, aj.), pomocí tohoto rozšíření pak představují skutečnost, kdy účastník má mnohem větší vliv na tvorbu mapy (komunikačního média) (Kubíček, Stachoň 2009). V tomto případě autoři zkoumají interaktivní kartografické vizualizace ve webovém prostředí, skrze prizmat využití map podle (MacEachren, Taylor 1994). V případě této práce se jedná o obdobné zhodnocení, kdy je však medium mapy, jakožto medium komunikace prostorové informace, „nahrazeno“ médiem virtuální reality, popř. virtuálním geografickým prostředím.

### 3.1.2 VR a kartografie

Kartografie v posledních 30 letech prošla tzv. "geoinformatizací." Geoinformatizace je přechod od papírových map k digitalizaci a automatizaci analytických i vizualizačních procesů. Digitalizace umožnila nové způsoby interakce a dynamické práce s prostorovými daty, a to prostřednictvím geografické informační systémy GIS. (Konečný 2011) Vývoj virtuální reality přirozeně zasáhl i geoinformační vědy. Zpočátku vedl k tvorbě Desktop VR vizualizacím, tedy zobrazování geoprostorové informace v 3D prostředí. Následně pak nastal posun do Immersive VR, kdy byl uživatel přímo vnořen do prostředí vytvořeného na základě geografických dat (Kvarda 2020).

Obecný přehled témat v kartografii využívajících 3D prostředí a různých úrovní jejich vizualizace poskytuje (Herman 2019). Patří mezi ně téma jako topografická data a analýzy nad nimi, 3D katastr, dokumentace a uchování kulturního dědictví, virtuální turismus, výuka geografie, orientace a navigace v zastavěných oblastech, vizualizace v hydrologii, podpora při krizovém řízení, vojenské simulace, geologie a geofyzika, meteorologie a teplotní, hlukové, ekologické a hlukové studie urbánních prostředí, modelování budov a facility management.

Virtuální prostředí v kartografii a geoinformatice jsou často spojovány s termíny VGE, *Digital Earth* a *Metaverse*. Následující podkapitoly se snaží tyto termíny objasnit a vysvětlit postoj této práce vůči nim.

#### VGE

Koncept VGE, zkratka pro virtuální geografická prostředí, představuje vývojový krok po dřívějších pojetích jako "virtuální geografie." Tento termín byl poprvé formálně navržen na počátku 21. století a od té doby prošel významným vývojem. Moderní popis VGE je

*Digitální geografické prostředí, které je generováno počítači a souvisejícími technologiemi, které uživatelům umožňují zažít a rozpoznat složité geografické systémy a provádět komplexní geografické analýzy. Toto*

*prostředí zahrnuje interakci mezi člověkem a počítačem, distribuované geografické modelování a simulace a síťovou geo-kolaboraci. (Chen, Lin 2018)*

VGE by mělo zrcadlit skutečné geografické prostředí ve virtuálním světě, a zároveň umožňovat uživatelům vizuálně vnímat a zkoumat analýzy ve virtuálním světě. VGE by tohoto měli dosáhnout kombinací tradičního geografického informačního systému (GIS) s technologiemi pro imerzní vizualizaci. (Çöltekin et al. 2020a)

VGE lze rozčlenit do čtyř hlavních komponentů (Lin, Chen, Lu 2013): **datový prostor, prostor pro modelování a simulace, prostor pro interakci a prostor pro spolupráci**. Data jsou v datovém prostoru organizována, manipulována a vizualizována, aby vytvořila digitální infrastrukturu pro VGE. Modelování a simulace jsou klíčovými prostředky pro moderní geografický výzkum. Interakce ve VGE vyžaduje odlišné myšlení než u desktopových počítačů, protože cílem je vytvořit zážitky srovnatelné se skutečným světem. Prostředí pro spolupráci umožňuje interakce mezi uživateli a je klíčem k řešení složitých geografických problémů pomocí spolupráce napříč různými oblastmi a odvětvími (Guo, Goodchild, Annoni 2020).

VGE je tedy možné chápat jako virtuální pracovny pro spolupráci na tvorbě analýz, vizualizací a následných interpretací nad geografickými daty. Tato práce bude zaměřena na určité aspekty datového prostoru, prostoru pro interakci a případě prostoru pro spolupráci. Cílem této práce není vytvořit kompletní VGE, ale představit možnosti a omezení současných technologií, které mohou být při budoucí tvorbě VGE využity.

### Digital Earth

Digital Earth (dále jen DE) je popisována jakožto virtuální atlas (*globe* – ve smyslu celé země) sestavený z masivních dat pozorování Země s více rozlišeními, více časovými obdobími, různými typy dat o Zemi a propojená s analytickými algoritmy a modely (Goodchild 2013; Grossner, Goodchild, Clarke 2008; Guo, Goodchild, Annoni 2020). Z vědeckého hlediska má DE dva základní aspekty. DE představuje rozsáhlý systém dat a informací, který agreguje a prezentuje data a informace týkající se Země a DE je virtuálním systémem Země, který může provádět simulace systému a podporu rozhodování pro složité geovědní procesy a socioekonomické jevy (Guo, Goodchild, Annoni 2020). Z hlediska VGE se jedná o další abstrakci nad konceptem VGE.

### Metaverse

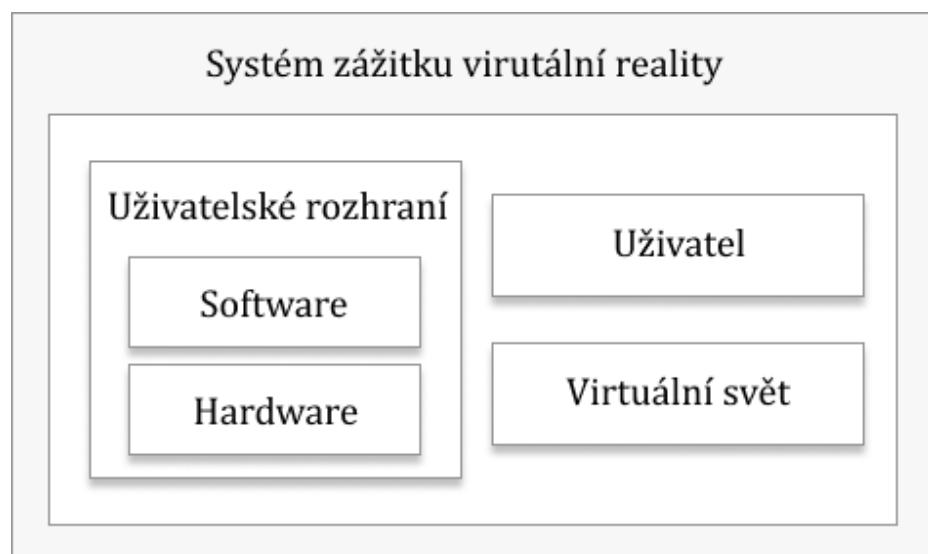
Podobně jako v případě definice virtuální reality, i koncept Metaverse je obtížně definovatelný a jeho význam se mění v závislosti na perspektivě a kontextu. Existuje mnoho různých názorů a přístupů k tomu, co Metaverse skutečně představuje. Jedním z aspektů, který ilustruje nejednoznačnost Metaverse, je různorodost definic a pohledů na tento koncept. Někteří ho vidí jako kolektivní termín pro digitální trojrozměrné světy, zatímco jiní vnímají Metaverse jako potenciální následující generaci internetu, která by mohla definovat Web 3.0 nebo se alespoň stát jeho součástí. Existují také představy o Metaverse jako o verzi internetu, kterým se člověk může procházet a „být v něm“ (Kiong 2022; Newton 2021). Dále je Metaverse často spojován s MMORPG počítačovými hrami, jelikož koncept individuálních virtuálních světů umožňujících sociální interakce odpovídá představám o Metaverse. Autoři (Ritterbusch, Teichmann 2023) na základě systematického průzkumu literatury vytvořili definici:

*Metaverse, spojení slov "meta" a "vesmír", popisuje (decentralizovaný) trojrozměrný online prostor, který je trvalý a imerzní, ve kterém se uživatelé zastoupení avatary mohou sociálně a ekonomicky účastnit, spolupracovat a být kreativní ve virtuálních prostorách oddělených od reálného fyzického světa.*

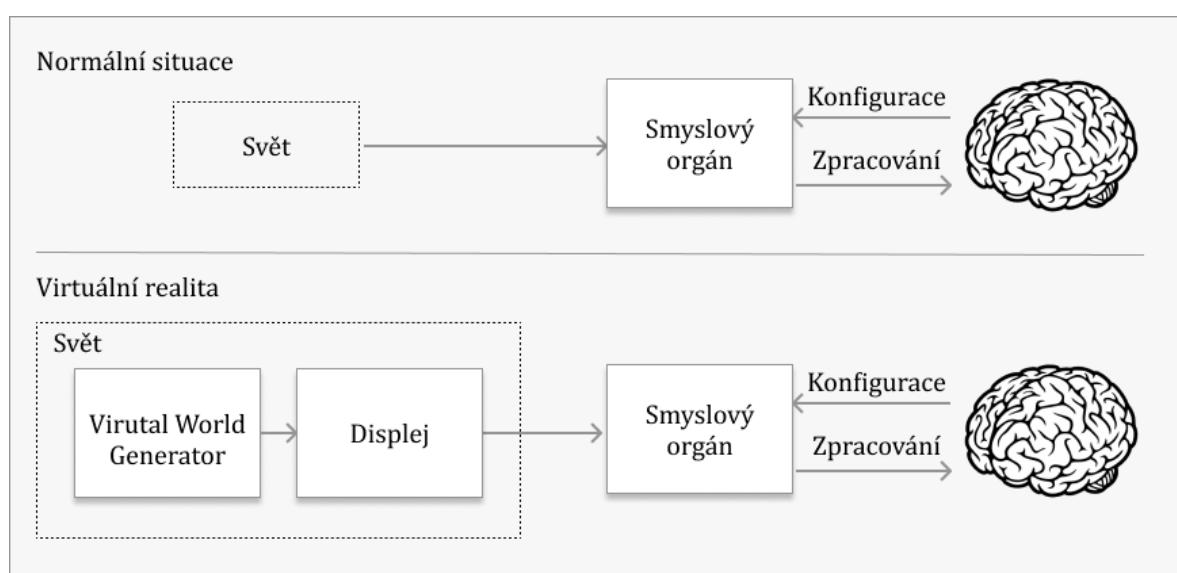
Všechny výše uvedené termíny (VGE, DE, Metaverse) se v různých měřítkách snaží definovat koncept virtuálního světa. Světa, který existuje paralelně se světem „reálným“ a je s ním oboustranně propojen, a to tak že z něj čerpá a zároveň jej zpětně ovlivňuje. Výše uvedené definice se zdají velice podobné a při pohledu z různých kontextů zaměnitelné. Pro tuto práci není důležité, zdali se výsledný virtuální svět jmeneje Digital Earth, Metaverse nebo VGE, ale zdali je možné a jakým způsobem k těmto vizím technologicky dospět.

### 3.2 Systém virtuální reality

Z obecného hlediska popisuje systém virtuální reality Sherman a Craig a dělí jej na základní komponenty (Software, Hardware, Virtuální svět, Uživatel) (viz. Obr. 2). LaValle rozděluje systém virtuální reality na obdobné komponenty (Hardware, Software a Uživatel). Toto rozdelení je patrné i v případě (Çöltekin et al. 2020b), kde autoři využívají rozdelení na (Technologie, Design a Uživatel). Kvarda definuje virtuální realitu jako systém kombinující počítačově generovaný svět s rozhraním uživatel-počítač (Kvarda 2020).



Obr. 2 Systém virtuální reality dle: upraveno dle: (Sherman, Craig 2019)



Obr. 3 Systém virtuální reality dle: upraveno dle: (LaValle 2020)

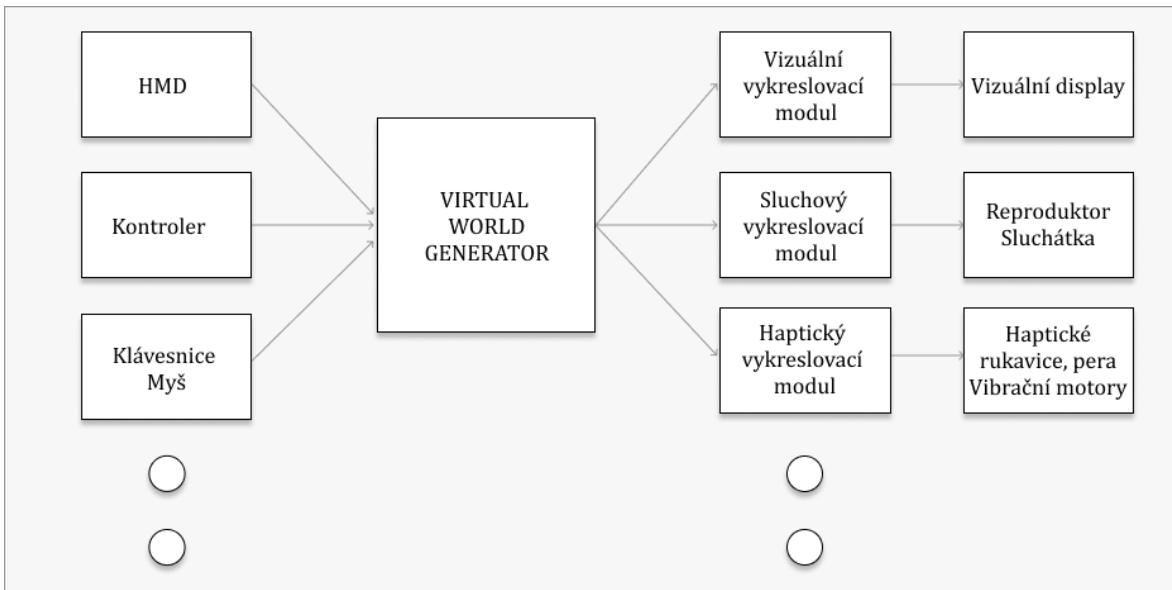
LaValle představuje koncept tzv. *virtual world generator*, čímž je myšlen komponent, který vytváří alternativní prostředí (může být VGE). Následně pak pomocí výstupu, který je specifický pro daný smyslový orgán, je percepce reálného světa nahrazena světem alternativním. Zmíněný VWG (*virtual world generator*) je tedy softwarová komponenta, která vytváří „jiný svět“, jímž může být kompletně syntetický svět, záznam existujícího světa a vše mezi tím. Člověk pak vnímá daný svět skrze jednotlivé smyslové orgány. Proces převodu VWG na display se nazývá *rendering*. Úspěšným VR systémem se pak rozumí takový systém, který je schopný do určité míry nahradit reálný svět světem vytvořeným. (LaValle 2020)

### 3.2.1 Hardware

LaValle dělí hardware na 3 kategorie a to: **displeje** (výstup), **senzory** (vstup) a **výpočetní jednotky** (*processing unit*). Výstupy je myšleno rozhraní počítač – člověk, přes které vnímáme virtuální svět. Nejdůležitějším je zrak, tedy rozhraní skrze obrazovku. V případě HMD se jedná o LED obrazovky s vysokým rozlišením nastavené blízko očí a zaostřeny skrze čočky. Klíčové pro vytvoření virtuálního zážitku je sledování polohy HMD v prostoru. Orientace je měřena pomocí IMU (*intertial measurement unit*), podrobněji pomocí gyroskopu a akcelometru. Další z klíčových vstupů sensorů je snímání okolí, části lidského těla nebo i celé postavy pomocí digitálních kamer a tzv. *depth* kamer využívající infračerveného záření. Mimo kamery pak součástí VR hardwaru jsou i mechanické vstupy jako klasické klávesnice, myš a jiné ovladače. Výpočetní jednotky se dělají primárně na mobilní HMD a *tethered* (externí počítač), kdy mobilní HMD jsou zpravidla méně výkonné.

### 3.2.2 Software

LaValle zmiňuje, že v ideálním případě by bylo vhodné, aby existoval nějaký specializovaný *VR engine*, díky němuž by vývoj VR systémů nemusel obsahovat nízko-úrovňové problémy (integrace HMD, aj.). Právě odhalení vhodné kombinace softwarových technologií pro tvorbu VR systému ve webovém prostředí s užitím geografických dat je jedním z cílů této práce viz. (kap. 4 Analyza technologií). Souhrn softwarových komponent potřebných pro vývoj virtuálního zážitku jsou zdůrazněny Obr. 4. VWG dostává input z nízko úrovňových systémů, které vyjadřují, co uživatel dělá v reálném světě. Jedná se o vstupní zařízení určující především polohu, orientaci (HMD) a následně interakci v prostoru (klávesnice, myš, ovladače). (LaValle 2020)



Obr. 4 Komponenty systému virtuální reality- upraveno dle: (LaValle 2020)

### 3.3 Komponenty systému virtuální reality

Tato kapitola podrobněji popisuje zmíněné hardwarové komponenty virtuální reality, rozdělené na výstupní a vstupní zařízení.

#### 3.3.1 Výstup - Output

Zobrazovací zařízení je možné klasifikovat dle míry imerze, kterou poskytují na ne-imerzní, semi-imerzní a plně-imerzní (Çöltekin et al. 2020b). Ne-imerzní by neměly být považovány za zobrazovací zařízení pro virtuální realitu. Za semi-imerzní jsou považovány často stereoskopické displeje, popř. monoskopické displeje, které zakrývají výraznou část zorného pole uživatele. Plně-imerzní jsou následně displeje, které kompletně vyplňují zorné pole uživatele. Plně imerzní v aktuální době jsou především HMD (*head mounted display*). Sherman a Craig dělí VR na 3 paradigmata, podle hardwaru, který zážitek zprostředkovává na: *hand based*, *stacionární* a *head mounted* (Sherman, Craig 2019). HMD zcela zaplňují zorné pole tím, že jsou obrazovky umístěny velice blízko očím. Lidské oko však není schopno ostřít na potřebně blízkou vzdálenost (cca 5–10 cm). HMD tedy využívají konvexní čočky položené do jejich fokální vzdálenosti. Využití konvexních čoček umožňuje, že obrazovka se jeví jako obrovský virtuální obraz, který je nekonečně daleko. (LaValle 2020).

V případě mobilních tradičních displejů je možné mluvit o „případné“ imerzi, kterou je možné dosáhnout kombinace s nástavci umožňující pohled na obrazovku skrze čočky a umístění displeje do vhodné vzdálenosti od očí. Tato skutečnost není validní v případě klasických displejů, které zakrývají pouze omezenou část zorného pole. Nutné však zmínit, že mobilní zařízení „vylepšené“ o nástavec na hlavu s čočkami se imerzí zdaleka nevyrovnaná speciálním HMD zařízením.

HMD systémy také musí počítat s různou délkou mezi zornicemi (*interpupillary distance – IPD*). Dalším z problémů je pak rozlišení obrazovky, které je potřeba aby nebylo možné rozeznat jednotlivé pixely. Na základě výpočtů založených na fyziologii lidského oka bylo zjištěno, že display by musel mít 16 000 x 16 000 px rozlišení na to, aby nebylo možné rozeznat jednotlivé pixely. Tento přístup je však velice výpočetně náročný tudíž, se aplikuje proces, kdy vysoké rozlišení je pouze tam kde je lidské oko aktuálně zaostřeno, k čemuž je však potřeba aby HMD podporoval *eye tracking* (LaValle 2020).

Výstupem není pouze vizuální, ale i haptický, popř. zvukový. Možnosti jednotlivých HMD se i těchto aspektech výrazně odlišují. V Tab. 1 jsou shrnutý klíčové charakteristiky populárních HMD. Pro vývoj aplikace je primárním kritériem, jaký způsob snímání, popř. interakce (*tracking*) implementuje (viz. kap. 3.3.2 Vstup – Input).

**Tab. 1 Přehled rozlišení populárních HMD. sestaveno dle: (Takle 2022; Mehrfard et al. 2019; Brown 2023)**

Náze HMD	Typ	Rok výroby	Rozlišení [na oko]	Obnovovací frekvence [Hz]	Tracking	Ovladače
Apple Vision Pro	Samostatný	Neznámý	3400x3400	90	6DoF - inside-out - oči - obličeje - ruce	x
HTC Vive Pro 2	PC - propojený	2021	2448x2448	120	6DoF - ouside-in	2x 6Dof - joystic, binární
HTC Vive	PC - propojený	2016	1080x1200	90	6DoF - ouside-in	2x 6Dof - joystic, binární
Lenovo Legion VR700	Samostatný	2022	1832x1920	90	6DoF - inside-out	2x 6Dof - joystic, binární
Meta Quest Pro	Samostatný	2022	1800x1920	90	6DoF - inside-out - oči, obličeje, ruce	2x 6Dof - joystic, binární
Oculus Quest 2	Samostatný	2020	1832x1920	120	6DoF - inside-out - ruce	2x - 6DoF - joystic, binární
Oculus Rift S	PC - propojený	2019	1280x1440	80	6DoF - inside-out	2x 6Dof - joystic, binární
Pico 4	Samostatný	2022	2160x2160	90	6DoF - inside-out - ruce	2x 6Dof - joystic, binární
Pimax Crystal QLED	Samostatný	2023	2880x2880	160	6DoF - inside-out - oči	2x 6Dof - joystic, binární
PlayStation VR2	PlayStation - propojený	2023	2000x2040	120	6DoF - inside-out - oči	2x 6Dof - joystic, binární
Valve Index	PC - propojený	2019	1440x1600	144	6DoF - ouside-in	2x 6Dof - joystic, binární
HP Reverb	PC - propojený	2020	2160x2160	90	6DoF - inside-out	2x 6Dof - joystic, binární

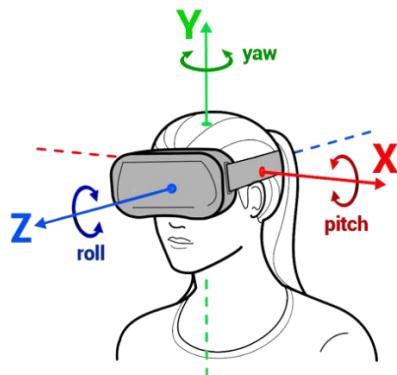
### 3.3.2 Vstup - Input

Důležitým aspektem hardware pro VR je *tracking* (Çöltekin et al. 2020b). Jedná se o získávání kontinuální informace o poloze a pohybu v rámci reálného světa. Díky vývoji mobilních telefonů a zmenšení a vylepšení Inerciálních měřících jednotek (IMU - *inertial measurement unit*) byl umožněn vývoj přesných metod pro snímání polohy (LaValle 2020). LaValle uvádí 3 hlavní kategorie toho co VR systém potřebuje snímat:

1. **Smyslové orgány uživatele** – převážně sledování pozice hlavy, popř. očí
2. **Ostatní části těla** – obličeje, ruce aj.
3. **Okolní prostředí** – reálné objekty v okolí uživatele

Důležitým konceptem pro *tracking* je koncept DoF (*degrees of freedom* – stupně volnosti). DoF je specifický způsob jakým se může pohybovat objekt v prostoru. Pohyby je možné zjednodušit na rotaci či posun po dané přímce. Jakýkoliv objekt se v prostoru může pohybovat maximálně pomocí 6 DoF, tedy translací a rotací kolem X, Y, Z os (Sherman, Craig 2019) viz. Obr. 7. Obecně 3 DoF znamenají rotace na daném místě a 6 DoF znamená rotace na daném místě a pohyb v prostoru (translace). Tracking pozice v prostoru je následně možné dělit na *inside-out* a *outside-in* přístupy. Jedná se o rozdělení na základě toho, jak zařízení snímá svoji polohu v prostoru. *Inside-out* snímá okolní prostředí kamery v rámci HMD, kdežto *outside-in* vyžaduje externí kamery, které snímají zařízení a tím určují jeho polohu. Návazně na výše zmíněné kategorie zobrazovacích zařízení je možné vytvořit dělení dle VR hardwarem poskytnutých DoF, které odpovídá i historickému vývoji těchto zařízení:

1. **3DoF** – Google Cardboard, Google Daydream, Samsung GearVR, Oculus Go atd. tedy „mobilní HMD“, kdy se jedná o obal s čočkami pro mobilní telefon. Produkty Daydream GearVR a Oculus Go poskytovali i ovladač, nejednalo se tedy pouze o stacionární HMD, ale byla zde možnost interakce. S nástupem HMD Oculus Quest se tyto produkty přestali vyrábět<sup>1</sup> (RoadToVR 2023).
2. **6DoF** – Oculus Quest 2, HTC Vive, PlayStation VR aj. umožňují snímání stacionární polohy i polohy v rámci prostoru pomocí metod zmíněných výše. Součástí těchto produktů jsou i ovladače, které také umožňují 6DoF. Tyto zařízení tedy umožňují rozšířenou interakci s virtuálním prostředím.



Obr. 5 Stupně volnosti – Degrees of Freedom – DoF (zdroj: (Dupin 2016))

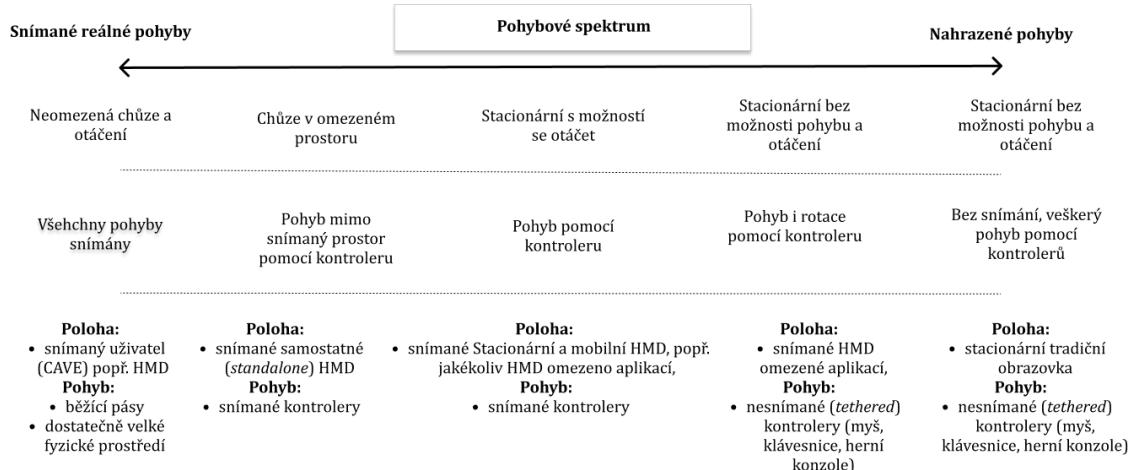
Stupně volnosti, které umožňuje vstupní zařízení je důležitým parametrem při vývoji VR aplikace, kdy je nutné v návrhu počítat s možnými variantami. O to víc pokud aplikace cílí na webové prostředí, kde cílovým uživatelem může být kdokoliv. Je nutné tedy počítat s kritérii:

1. Kolik DoF aplikace bude podporovat, aneb jaké možnosti pohybu uživatel má.
2. Zda HMD je pouze display nebo jsou k němu přiřazeny i dodatečné ovladače. Následně pak kolik DoF tyto ovladače mají (2 – klávesnice a myš, 3 – rotační, 6 – rotační i poziční).
3. Zda výpočetní jednotka je stolní počítač, mobilní telefon, popř. samotný HMD.

Pro tvorbu virtuálních prostředí je nutné zmínit problematiku **lokomoce**, tedy pohybu ve virtuálním a reálném prostředí. Ve většině případů při tvorbě VR je virtuální svět, ve kterém má uživatel umožněn pohyb větší než fyzické prostředí, ve kterém se nachází. Z toho důvodu

<sup>1</sup> Zastavení výroby produktů znamená i ukončení podpory ve vybraných softwarových řešeních viz. (dmarcos 2023)

musí dojít při přenosu pohybů (translace a rotace) k nahrazení (*remapping*) pohybů z fyzického světa do světa virtuálního pomocí náhradních vstupů (např. pohyb joystickem na ovladači) namísto pouhého snímání reálné polohy. La Valle poskytuje vysvětlení ve formě pohybového spektra, které vyjadřuje, do jaké míry jsou pohyb a poloha v reálném světe snímány a projektovány, popř. nahrazeny ve světě virtuálním viz. Obr. 6. Při řešení nahrazení pohybu je nutné brát v potaz problematiku neshody reálného pohybu a jeho nahrazení pomocí vizuálního vjemu, který může vést k tzv kinetóze. (viz. kap. 3.4.2 Percepce pohybu).



Obr. 6 Pohybový spektrum a lokalizace typů vstupních a výstupních zařízení v jeho rámci. upraveno podle: (LaValle 2020)

Kategorizaci způsobů lokomoce rozvádí (Boletsis 2017), který uvádí kategorie:

- **Pohybové** – Pohyb v reálném světe je promítán do virtuálního, v případě většího VP jsou pak implementovány techniky chůze na místě, pohyb rukou aj.
- **Omezené** – Velikost virtuálního prostředí je definována překážkami v reálném prostředí, pohyb je tedy mapován 1:1.
- **Pomocí ovladačů** – Pohyb je realizován pomocí ovladačů, a to pomocí joysticků, tlačítek, popř. pomocí náklonů hlavy.
- **Teleportací** – Realizuje pohyb ve VP nekontinuálně, uživatel je teleportován na vyznačené místo instantně, inicializace teleportace je často implementována skrze tlačítka ovladačů.

Interakce s virtuálním prostředím nespočívá však pouze v pohybu uživatele prostředím, ale i interakce s objekty. Interakce s virtuálními objekty často probíhá skrze procesy **selekcii**, **manipulace** a **umístění** objektů, tedy způsoby pohybu (translace, rotace a transformace) virtuálních objektů. Druh vstupní informace lze následně rozdělit na (Menard 2019):

- **Metrické** – pohyby jsou snímány v prostoru (různé úrovně DoF – myš: 2, HMD kontrolér: 6, snímání rukou: 6 atd.), popř. joystick umístěný na kontroléru.
- **Binární** – stlačení tlačítka. V případě webových technologií kategorizaci a standardizaci práce s různými vstupy zprostředkovává WebXR API (viz. kap. **Error! Reference source not found. Error! Reference source not found.**). Dle tohoto API lze vstupy rozdělit na dvě hlavní kategorie, jimž jsou (MDN Contributors 2023a):
  - **Zaměření (targeting)** – Specifikace bodu ve virtuálním prostoru uživatelským vstupem, tedy dotykem obrazovky, sledování očí, popř. použití joysticku.)

- **Akce (action)** – Jedná se o stlačení tlačítka, popř. jiná binární operace.

Dělení dle míry interakce popisuje (Bořil 2022) ve své diplomové práci. Interakce může být nulová, kdy uživatel pouze sleduje virtuální prostředí. Další úrovní možností interakce ve VP je pomocí pohledu (*gaze based*), použitelný v případě HMD bez ovladačů, kdy ve směru pohledu uživatele je vysílán paprsek, pomocí kterého je možné interagovat s VP. Interakce (akce) je zpravidla realizována pomocí časového prodlení na daném objektu, popř. tlačítkem. V případě existence ovladačů, je zdroj paprsku vysílán z ovladače a interakce spouštěna pomocí tlačítek (*pointer based*). Za účelem zahrnutí zařízení bez HMD ovladačů je vhodné v aplikaci umožnit interakci pomocí tradičních vstupních zařízení jako je myš a klávesnice (Menard 2019). Způsoby interakce ve VP jsou ovlivněny tím, že uživatel není schopen vidět reálné vstupní zařízení (např. polohu tlačítka na klávesnici). Řešením je převod klávesnice, ovladačů aj. do virtuálního prostředí v podobě virtuálních modelů nebo i vstup pomocí snímání pohybů rukou popř. ovládaní pomocí hlasu (Vicente et al. 2022).



Obr. 7 Vybrané příklady řešení interakce ve VP. zdroj: (Mozzila Corporation 2023a; Ravasz 2019)

Typ interakce s aplikací je klíčový v návrhové části, jelikož je nutné definovat jaké typy interakce bude podporovat, což je přímo spojené s rozhodnutím, jaké vstupní zařízení bude podporováno. Obecně při návrhu aplikace je volba mezi alternativami:

1. 3DOF HMD a žádným input
2. 3DOF HMD a 3DOF input.
3. 6DOF HMD a 6DOF input.

V případě č.1 je obecným přístupem implementovat *gazed based* input. Kdy uživatel interahuje s uživatelským rozhraním, popř. celkově s objekty ve scéně pomocí paprsku ve směru pohledu. Interakce je uživateli komunikována např. skrze časovač, který ukazuje interval prodlení, než bude interakce vykonána. V druhém případě je uživatel schopen zaměřit v prostoru nezávisle na pohledu. Třetí variantou je plná podpora pro pohyb uživatele pomocí ovladačů. Využití potenciálu takovéto míry interakce vyžaduje často implementaci pokročilého pohybu (teleporatace) a interakce s objekty (translace, rotace) popř. přímá manipulace (chycení objektu). (three.js Contributors 2023f) Zároveň je vhodné zahrnout i podporu pro neimerzní zařízení, tedy podporu pro pohyb a interakci s 3D scénou pomocí tradičních vstupních zařízení (klávesnice, myš).

## 3.4 Percepce

Nejdůležitějším komponentem v rámci systému zážitku virtuální reality je účastník / uživatel, proto je nutné rozumět procesu vnímání (percepce) (Sherman, Craig 2019). Za účelem tvorby VR prostředí / zážitku je nutné znát způsob jakým lidský mozek interpretuje předanou informaci skrze smyslové orgány (LaValle 2020). V případě percepce je možné hovořit o počítcích ( inicializované podněty ze smyslových orgánů). Více počítek tvoří vjem. (Chloupková 2007) Pro VR je klíčová percepce (vnímání) vzdálenosti a měřítka, pohybu, barvy a následně jejich kombinace (LaValle 2020).

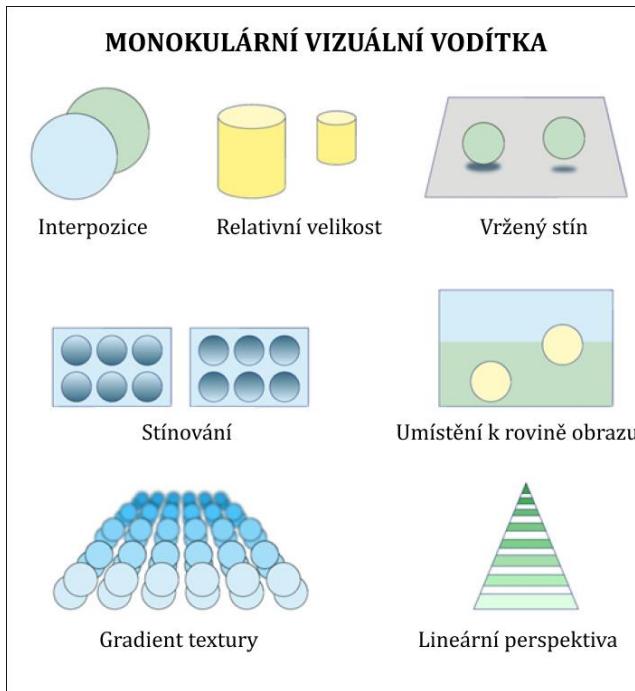
### 3.4.1 Percepce vzdálenosti

Percepce vzdálenosti má za úkol dle (Mather 2016) vyřešit 4 hlavní problémy jimiž jsou:

1. Stanovení pořadí objektů v prostoru
2. Stanovení intervalů mezi objekty
3. Stanovení absolutní vzdálenosti objektu od pozorovatele
4. Stanovení odhadu trojrozměrného povrchu a jeho tvaru

Za účelem dosažení těchto cílů využívá určitá vodítka, jelikož obraz na sítnici je dvojrozměrný. Vodítka je možné dělit na monokulární a binokulární na základě toho, zda jsou vnímána jedním či oběma očima. Dále je pak možné vodítka rozdělit na statická a dynamická. Při vnímání jsou tato vodítka kombinována. (Mather 2016; Chloupková 2007). Monokulární vodítka (viz. Obr.X) (Mather 2016; Chloupková 2007; Matatko, Bollmann, Müller 2011):

- **Interpozice** – překrývaný objekt je vnímán jako vzdálenější
- **Relativní velikost** – menší objekty jsou považovány za vzdálenější,
- **Vržený stín** – objekt vytváří stín na jiném povrchu / objektu
- **Stínování** – tvar objektu vytváří stín na sobě samém
- **Umístění v rovině obrazu** – objekt blíže horizontu je považován za vzdálenější
- **Gradient textury** – textura je s rostoucí vzdáleností hustší a jednotlivé prvky jí tvořící jsou menší



Obr. 8 Monokulární prostorová vodítka (upraveno dle: (Bogdanova, Boulanger, Zheng 2016))

Dynamická vodítka jsou **paralaxa pohybu** – pohyb vzdálenějších objektů v prostoru je po sítnici oka pomalejší (Chloupková 2007), **mizení (deletion)** – vzdálenější objekt „mizí“ za bližší, a **přirůstání (accretion)** – vzdálenější objekt „přirůstá“ není zakryván objektem bližším. (Matatko, Bollmann, Müller 2011).

Mezi binokulární vodítka se pak řadí: **binokulární disparita** – rozdílná poloha očí umožňuje vidět prostor z jiného úhlu, což dodává možnost vytvořit percepci vzdálenosti a **binokulární konvergence** – jedná se o vjem pohybu očí, kdy s blížícím se objektem se oči stáčí k sobě a se vzdalujícím naopak, mozek tedy tyto pohyby interpretuje jako změnu ve vzdálenosti objektu. (Chloupková 2007) Binokulární disparita a konvergence jsou základní procesem při vnímání VP skrze HMD. HMD imituje stereoskopické vidění pomocí dvou nezávislých displejů, na nichž zobrazovaný obraz je mírně posunut, za účelem vytvoření iluze hloubky.

### 3.4.2 Percepce pohybu

Percepce pohybu velice výrazně závisí na vizuálním vjemu. Většina soudobých HMD se soustředí primárně na vizuální vjem, tudíž neshody ve virtuálním prostředí mohou vést k nekorektním vjemům pohybu, což může vést k nevolnosti. Problém pro VR systémy tvoří iluze vlastního pohybu z důvodu vnímání pohybu vizuálně. Jedná se tedy o konflikt visuálního a rovnovážného aparátu nazvaný *vection*. Pokud tento konflikt vede k nevolnosti je možné mluvit o *kinetóze*. Problém se vyskytuje často v případě akcelerace pohybu avatara ve virtuálním prostředí. V případě, kdy akcelerace je postupná dochází v mozku k detekování neshody ve vizuálních a rovnovážných (střední ucho) vjemech. Absolutní akcelerace / teleportace v transpozici i v rotaci je řešením pro tento problém, jelikož mozek vyhodnotí nárůst rychlosti jako extrémní případ a nestihne spustit reakci na něj (bolest hlavy, nevolnost) (LaValle 2020).

*Retinal image slip* aneb problém kolik FPS (*frames per second*) je dostatečné pro VR display. VR display vyžaduje vyšší hodnoty FPS, jelikož je nutné vhodně upravit pohyb pozorovaného objektu ve virtuálním světě tak aby zůstal zaostřený i při pohybu hlavy. Je nutné zachovat tento

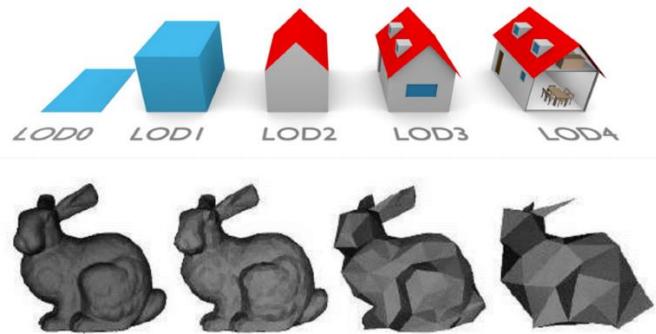
objekt v jednom bodě na sítnici, tudíž v rámci virtuálního světa je nutné, aby se tento fixovaný objekt posunul po obrazovce opačně od pohybu hlavy. Z důvodu nedostatečné hodnoty FPS je obraz objektu na obrazovce příliš dlouho a uživateli se pak jeví jako přeskakující („*judder*“) namísto plynule se pohybujícího. Jelikož vysoké FPS hodnoty jsou zatím nevyřešený výrobní problém je *retinal image slip* řešen skrze *low persistence* přístup. Kdy je obraz objektu zobrazen jen v dané intervaly, které stačí očním receptorům na zaznamenání obrazu. Pro příklad moderní HMD Oculus Quest 2 umožňuje hodnoty FPS až 120Hz. (LaValle 2020)

V případě, že nejsou všechny smyslové vjemy nahrazeny virtuálními vstupy nebo pokud vstupy nejsou dokonalé (jsou v konfliktu s lidskou fyziologií) dochází ke konfliktům vnímání. Nejvíce problematickým se dlouhodobě jeví vekce popř. kinetóza tedy konflikt mezi vizuálními a rovnovážnými vjemy. Dalším z častých problémů je konflikt ve výšce uživatele (např. v sedě) a avatara v rámci virtuálního světa. Navíc k nesouladům mezi smysly přispívají i nedokonalosti v hardware, software, obsahu a rozhraních VR, což způsobuje nesoulad s reálnými zkušenostmi. Tyto konflikty pak vedou ke špatným či nechtěným interpretacím, popř. k nevolnosti a únavě. (LaValle 2020)

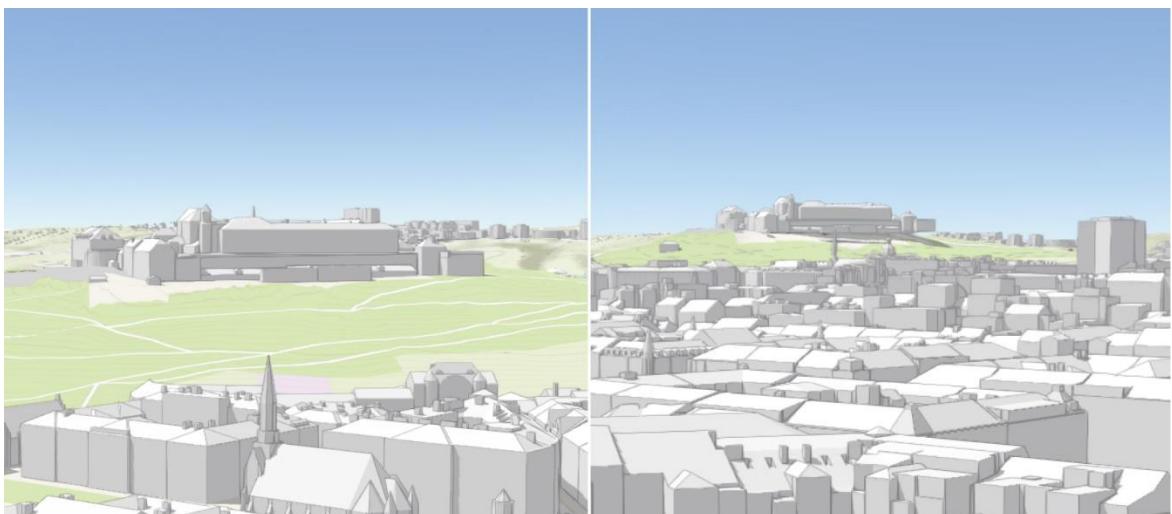
## 3.5 3D modelování koncepty a principy

### 3.5.1 LOD

LOD (*level of detail*) je široce využívaná technika v oblasti počítačové grafiky, GIS a geoprostorové vědy. Cílem je dosáhnout vyváženého poměru mezi vizuální kvalitou a výkonem prostřednictvím přizpůsobení úrovně detailu na základě vzdálenosti objektu nebo scény od pozorovatele. V počítačové grafice se LOD používá k optimalizaci renderování pomocí dynamického přepínání mezi různými reprezentacemi objektu nebo scény (viz. Obr. 9 Příklad konceptu sémantického LOD v případě specifikace CityGML – nahoře; v případě redukce vertexů – dole. (převzato z: (Biljecki, Ledoux, Stoter 2016) a (Ghulam et al. 2013)). S rostoucí vzdáleností pozorovatele od objektu se používají verze s nižším stupněm detailu, což snižuje výpočetní nároky a zlepšuje rychlosť renderování. Nižší stupně detailu mohou být dosaženy sémantickým zjednodušením (3D modely měst – CityGML) nebo algoritmickým (menší počet vertexů). Tímto je umožněno *real-time* renderování složitých scén nebo velkých datových sad, což jsou klíčové požadavky pro tvorbu úspěšné virtuální reality. V GIS techniky LOD pomáhají spravovat tato data prostřednictvím automatického přizpůsobování úrovně detailu na základě prostorového kontextu nebo úrovně přiblížení uživatele. V případě prostorových věd je pak možné mluvit o generalizaci. V některých případech však tento přístup není zcela vhodný viz. (Obr. 10) z nějž je patrné, že aplikovaný LOD přístup na data DMT vede k tomu že budovy (hrad Špilberk) jsou při pohledu z dostatečné vzdálenosti nevhodně zobrazeny nad zjednodušeným terénem. Případně řešení tohoto problému pak poskytuje práce (Semmo, Döllner 2014), kde autoři navrhují interaktivní změnu LOA dle akcí uživatele.



Obr. 9 Příklad konceptu sémantického LOD v případě specifikace CityGML – nahoře; v případě redukce vertexů – dole. (převzato z: (Biljecki, Ledoux, Stoter 2016) a (Ghulam et al. 2013))



Obr. 10 Snímky obrazovky z aplikace 3D model města Brna. Hrad Špilberk pohledu ze směru Vila Tugendhat – Špilberk. vlevo – terén je vykreslován v plném rozlišení; vpravo – terén je vykreslován zjednodušeně, model hradu je posazen nad terénem. Zdroj dat: (KAM Brno 2023)

### 3.5.2 Problém měřítka

V kontextu kartografie je problematika v konfliktu s konceptem měřítka, jelikož v tradiční 2D kartografii je jednoznačně možné říct jaké měřítko jaká vizualizace (mapa) má, kdežto v případě 3D map či virtuálních prostředí se odborná literatura není schopná shodnout zdali je koncept měřítka aplikovatelný, jelikož je nutné brát v potaz zobrazené území, LOD, míru přiblížení a následně tedy i jejich změnu v případě interakce. (Bandrova, Bonchev 2013)

Ve virtuálním prostředí na rozdíl od tradičních map je měřítko dynamické, uživatel je tedy případně schopen interagovat s daty v libovolném měřítku. Dynamické měřítko umožňují i webové mapy. Webové mapy definují pro možná měřítka hranice. Vnější hranice, tedy minimální měřítko je často určena tím, že v tomto měřítku je možné kompletně obsáhnout mapované území. Maximální měřítko je určeno na základě podrobnosti dat, tedy jaká nejpodrobnější data je možné zobrazit. Tato hranice by měla končit na maximálním měřítku v jakém byla mapa / data pořízeny. Například při vizualizaci rastrové mapy pořízené v měřítku 1:5000. Webové mapy dosahují dynamického měřítka pomocí vykreslování mapy v daném měřítku na základě míry přiblížení (virtuálně vypočtené měřítko pro obrazovku). Technologicky je toto dosaženo pomocí dlážic (MapTiler, Mapy.cz, Mapbox atd.), popř. vykreslování map ve vhodném měřítku pro určené intervaly přiblížení (CUZK) (viz. Obr. 11)

Ve virtuálním prostředí je však koncept měřítka složitější a je spekulativní, zda je vůbec aplikovatelný. Ve 3D prostředí se vykreslování provádí skrze perspektivní pohled, tudíž není

zcela jasné, k čemu měřítko vztahovat. Zároveň ve virtuálním prostředí je běžný pohyb ve všech osách. Obdobný přístup jako pro 2D webové mapy je umožněn pomocí 3D objemových dláždíc. Data jsou rozdělena do virtuálních krychlí, které jsou zobrazované uživateli v daném rozlišení na základě jeho vzdálenosti. Dynamické měřítko je tedy řešeno dynamickými daty.

V případě tvorby virtuální aplikace na základě dat statických, tedy „ne-dlaždicových“, je nutné vzít v potaz jaký typ v jakém měřítku dává smysl popř. je možné zobrazovat ve virtuálním prostředí. Toto rozhodnutí je klíčové, jelikož virtuální realita umožnuje zobrazení dat v libovolných měřítkách. Rozhodování je možné rozdělit na základě tradičního dělení kartografické tvorby tedy **Topografické a Tematické mapy** (Kraak, Ormeling 2020).

### **Topografické mapy:**

Zpravidla topografické mapy zobrazují území tak jak existuje v realitě, tedy zobrazují tvar povrchu, vegetaci, vodstvo, urbánní prostředí (zastavěné plochy, komunikace atd.).

- **Velká měřítka (1:1 – 1:5 000):**

- **Měřítko pořízení** – Největší měřítko, v jakém jsou tyto zdroje poskytovány je 1:5000. Pro větší měřítka následně chybí informace a jak je patrné z Obr. 11 dosažení větších měřítek i v případě 2D map z toho důvodu vede ke ztrátě kvality vizualizace.
- **Technická omezení** – Zobrazení ve velkých měřítkách ve 3D prostředí vyžaduje vysokou podrobnost geometrie a detailní textury. Míra podrobnosti je však rázně omezena hranicemi možností technologií umožňující VR (viz. kap. 3.6.4 Výkon)
- **Práce s LOR (level of realism)** – Za účelem vyřešení výše zmíněných problémů je možná práce s nižším LOR, tedy volba schematické vizualizace.

- **Střední měřítka (1:5000 – 1:1 000 000):**

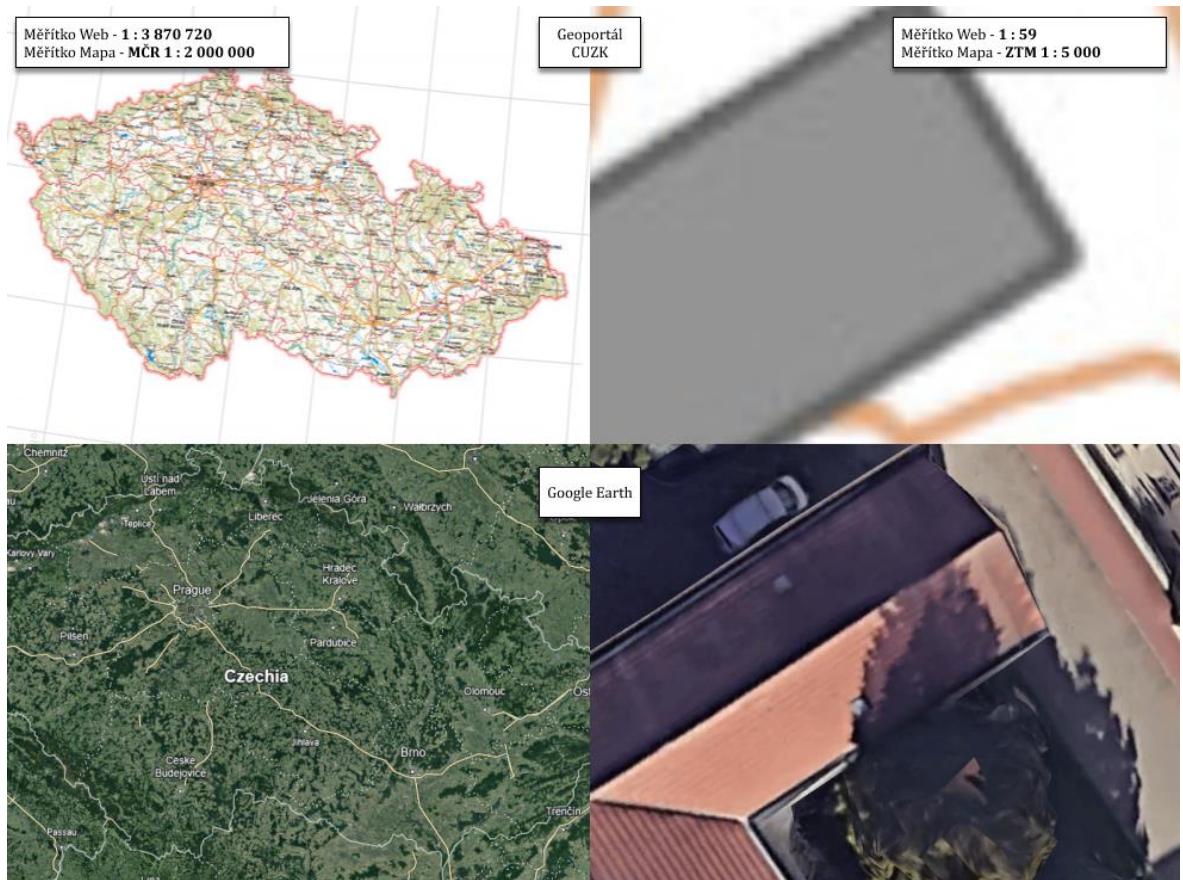
- **Zobrazitelný obsah** – Ve virtuální realitě je klíčovým prvkem možnost zobrazit data ve 3 dimenzích. V případě topografických map třetí dimenze znamená nadmořskou výšku. Je tedy nutné vzít v potaz rozsah hodnot nadmořské výšky zobrazovaných dat v daném měřítku, tak aby relativní rozdíl např. ve výšce terénu nebyl příliš malý (možné řešení skrze různé hodnoty měřítka v různých osách – zvýraznění nadm. výšky).

- **Malá měřítka (1:1 000 000 – 1:100 000 000):**

- **Zobrazitelný obsah** – V těchto měřítcích přetrvává problém malých relativních hodnot nadmořské výšky. V případě vizualizace celého světa se pak jedná o virtuální glóby, kdy nadmořská výška je vůči velikosti země mizivá. Řešení je tedy obdobné jako využívají reálné glóby se zvýrazněným reliéfem.

### **Tematické mapy**

Tematické mapy umožňují mnohem větší volnost ve volbě měřítka na rozdíl od topografických. Tematická data jsou zpravidla dostupná napříč měřítky v různých seskupení (budovy, admin. Jednotky, státy, kontinenty atd.). V případě tematických map je využití 3. dimenze na autorovi vizualizace, jelikož je možné pomocí 3. dimenze zobrazovat libovolné charakteristiky.



Obr. 11 Porovnání měřítek map ve webovém prostředí ve 2D prostředí a 3D prostředí. Zmenšeno – měřítka odpovídají zobrazení v rámci internetového prohlížeče namísto velikosti obrázku v textu. Zdroj dat: (ČÚZK 2023a)

Jako důležitou problematiku zmiňuje (Çöltekin et al. 2020b) míru abstrakce v návrhu virtuálních prostředí. Obecně je snaha dosáhnout co nejvíce realistické vizualizace. Ta je však nutná překonat problémy náročnosti zpracování množství dat a jeho vykreslení za předpokladu udržení nízké latence, vysokých hodnot FPS a dostatečného rozlišení. Díky této problematice je nutné využívat LOD a LOR přístupy (Çöltekin et al. 2020b). Při zanedbání některých z těchto aspektů v návrhu vizualizace se pak jedná o snížení míry imerze. Další z problematik je pak tvorba virtuálního obsahu. Vytvoření obsahu s vysokým LOD a LOR zahrnuje řadu komplexních operací (Çöltekin et al. 2020b). Za hlavní úskalí se považuje kombinace množství vstupních dat (LiDAR, tomografie, stereofotogrammetrie aj.), manuální práce při samotném modelování jako např. segmentace textur od povrchů, tvorba topologicky správný povrchů, fyzikální simulace aj. Z hlediska této práce je klíčový výběr vstupních dat uskutečnit tak, aby potřeba výše zmíněných kroků byla minimalizována, popř. ošetřena již existujícími řešeními.

### 3.5.3 Datové modely a formáty

Pokud má virtuální realita zobrazovat realitu skutečnou, a to v různých úrovních abstrakce, je pravidlem, že data 3D scén jsou často velice obsáhlá. V případě, kdy se vizualizace pohybuje v prostředí internetu je velikost, a hlavně rychlosť načítání dat významným faktorem. Je proto nutné pečlivě vybrat datový formát. Kritéria výběru formátu závisí primárně na podpoře ve webovém prostředí, možnosti přenosu geografických dat, interoperabilitě s jinými formáty (např. zdali je přístupná transformace z jiných vstupních formátů), zda se jedná o proprietární

či otevřený formát, k jakému účelu je formát primárně určen a jak moc je formát aktuální (do jaké míry je využíván).

### glTF

Formát **gltf** přezdívaný *JPG pro 3D*. Jedná se o otevřený formát vytvořený skupinou Khronos. Formát má primárně dvě formy – jakožto binární balík *.glb* nebo jako JSON soubor *.gltf* indexující připojené binární soubory (atributy - *.bin*, textury - *.jpg*, *.png*, *.webP*). Formát je podporovaný ve většině WebGL knihoven, grafických softwarů a nástrojů pro konverzi, kompresi atd. Primárním zaměřením glTF je schopnost obsahnout kompletní 3D scénu se všemi komponenty (viz. kap. 3.6.2 Komponenty 3D vizualizace).

Geografická data jsou často velice obsáhlá a vyžadují velká množství geometrie pro adekvátní vyobrazení. Dále vyžadují uchování vztahu atributových dat (metadat) a jim přiřazené geometrie a zápis souřadnicových systémů. Z tohoto důvodu je ve vývoji tzv. geoprostorový profil. Jedná se o standardizovaný způsob, jak umožnit glTF streamování obsáhlých terénních dat s texturami, bodových mračen a CAD modelů a aby bylo možné propojit metadata (atributy) s geometrií.

Pro velké objemy primárně geografických dat v 3D scénách vytvořila společnost Cesium formát 3D Tiles. Jedná se o otevřený formát a OGC standard. Formát je vytvořen na základě glTF specifikace. Hlavní předností je hierarchický LOD přístup, kdy data jsou definována ve stromové struktuře, kde koncové nody mají maximální rozlišení a každý rodič je zjednodušenou verzí svých potomků. Tento přístup tedy umožňuje streamovat pouze data potřebná pro dané zobrazení. 3DTiles formát sám o sobě určuje způsob jakým rozdělit 3D model do objemových dlaždic, kdy každá dlaždice odkazuje na glTF. V rámci glTF je pak zakódována samotná geometrie, textury, komprese, identifikace prvku a metadata. (Geospatial Webinar 2023)

Obdobným formátem jako 3D Tiles je I3S/SLPK (*Indexed 3D Scene Layers*), formát podporuje sdílení 3D objektů, povrchových sítí s texturami, bodová mračna aj. Formát primárně vytvořen a využíván v Esri technologiích a zároveň OGC standard. (OGC 2023)

Mezi nejpopulárnější formáty pro výměnu 3D dat se také řadí FBX, OBJ a DAE, každý z těchto formátů však má určité nedostatky, které zabránily rozsáhlé adopci napříč softwary. OBJ neumožňuje ukládání animace, FBX je proprietární formát a DAE má příliš komplexní strukturu (Discover three.js Contributors 2023). Žádný z těchto formátů na rozdíl od glTF nemá aspirace pro podporu geografických dat. Z těchto důvodů bude v rámci této práce primárním výmenným formátem glTF.

V případě tvorby 3D prostředí je nutné zmínit také možné formáty pro textury, jelikož standardním postupem dosažení vizuálního realismu při tvorbě VP je využití bitmap pro určení vzhledu geometrie. V kontextu virtuální tvorby je možné hovořit o 3 hlavních kategoriích.

- **Běžné formáty:** PNG, JPG
- **Webové formáty:** WEBP
- **Komprimované / Optimalizované:** .basis, KTX2

Tyto formáty je nutné využívat na základě účelu 3D modelů. PNG formát je vhodný při převodu modelů mezi formáty, jelikož je bezzáratový a široce podporován. WEBP je vhodný v případě malých až středních scén, kdy není výrazný důraz na výkon. KTX2 je optimalizovaný formát minimalizující GPU paměť. KTX2 je tedy vhodný v případě prostředí obsahující velké množství modelů vyžadujících detailní textury. V kontextu geoprostorových informací je toto aktuální problém, protože při prezentování geografické informace (terén, objekty) v měřítku

1:1, měly by být textury dostatečně detailní. GLTF podporuje všechny výše zmíněné formáty. (Chadwick 2023)

Formáty lze dále dělit na základě druhu dat pro který jsou primárně určeny. Zpravidla se jedná o mračna bodů (PCD, EPT), 3D modely (3DS, OBJ), 3D scény (Collada, glTF, KML), modely měst (CityGML, City JSON) (Mezzo 2019). Dále je možné dělit formáty dle notace, otevřenosti a využití v rámci webového ekosystému.

**Tab. 2 Seznam relevantních datových formátů umožňující tvorbu 3D vizualizace. (EduTech Contributors 2023)**

Formát dat	Použití	Notace	Otevřenost	Web
3D Tiles	Web GIS	JSON / Binární	Open Source	Ano
i3s (I3S Scene Layer)	Web GIS	JSON / Binární	Open Source	Ano
slpk (Scene Layer Package)	Web GIS	Binární	Open Source	Ano
KML (Keyhole Markup Language)	Web GIS	XML	Open Source	Ano
GeoJSON	Web GIS	JSON	Open Source	Ano
CityJSON	WebGIS / Urbanismus	JSON	Open Source	Ano
CityGML	Urbanismus	XML	Open Source	Ano
IFC (Industry Foundation Classes)	Modelování staveb (BIM)	Binární	Open Source	Ne
Shapefile - Polygon Z / Multipatch	GIS	Binární	Proprietární (ESRI)	Ne
DWG (AutoCAD)	CAD	Binární	Proprietární (AutoDesk)	Ne
Collada	3D Grafika a VR	XML	Open Source	Ano
OBJ (Wavefront)	3D Grafika a VR	Text	Open Source	Ano
glTF (GL Transmission Format)	3D Grafika a VR	JSON, Binární (glb)	Open Source	Ano
X3D	3D Grafika a VR	XML	Open Source	Ano
VRML (Virtual Reality Modeling Language)	3D Grafika a VR	Text (VRML), XML (X3D)	Open Source (X3D)	Ne
netCDF (Network Common Data Form)	Vědecká data	Binární	Open Source	Ne
HDF5 (Hierarchical Data Format)	Vědecká data	Binární	Open Source	Ne
FBX (Filmbox)	3D Modelování a Animace	Binární	Proprietární (Autodesk)	Ne
PLY (Polygon File Format)	3D Grafika a Vizualizace	Binární	Open Source	Ano

Tab. 3 Formáty pro textury. (Chadwick 2023)

Formáty	Komprese	Velikost (disk)	Velikost (GPU)	Alpha	gltf podpora	Charakteristika
PNG	Bezstrátová	Vysoká	Vysoká	Ano	Ano	Jednoduché modely, Vysoká kvalita, Výborná kompatibilita
JPG	Ztrátová	Malá	Vysoká	Ne	Ano	Jednoduché modely, Malá velikost, Výborná kompatibilita
WEBP	Vysoce ztrátová	Velice malá	Vysoká	Ano	plugin	Jednoduché modely, Malá velikost
KTX2	Ztrátová	Střední	Nízká (komprimace)	Ano	plugin	Komplexní modely, Rychlé GPU načítání,

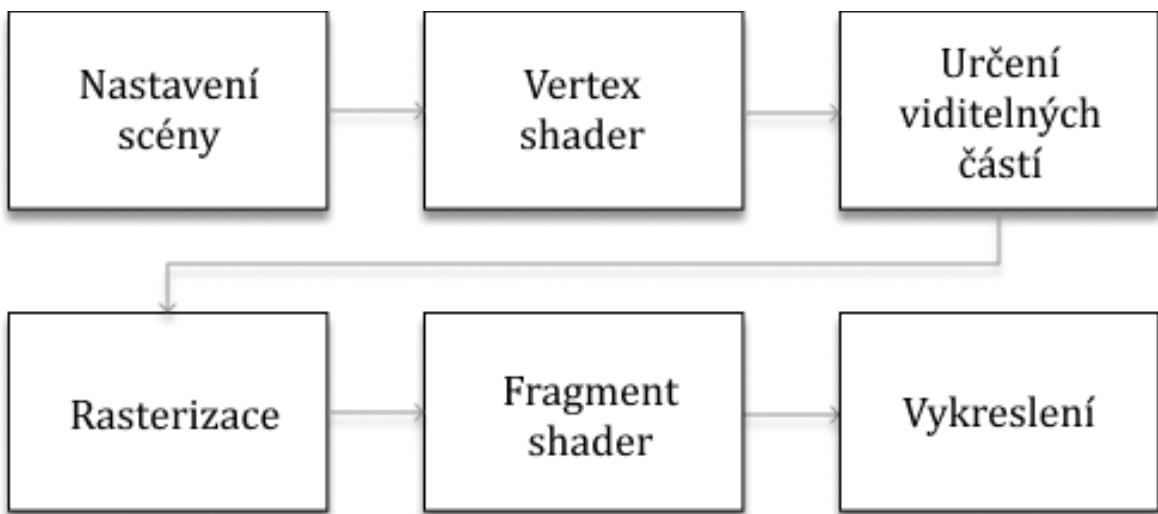
## 3.6 3D grafika

### 3.6.1 Rendering pipeline

Za účelem úspěšné práce s 3D grafikou je nutné mít základní přehled o procesech, které umožňují vykreslování 3D prostředí na 2D obrazovku. Tento proces se v oblasti počítačové grafiky nazývá *graphics rendering pipeline*. Proces se skládá z několika etap, pořadí a specifikace etap se různí napříč implementacemi (Dunn, Parberry 2011). Následující popis je tedy velice obecný. Rozdělení etap v případě WebGL (viz. **Error! Reference source not found.**) je následující: (Ghayour, Cantor 2018; Ariën 2017; Sherif 2018; Dunn, Parberry 2011):

1. **Nastavení scény:** Základním nastavením je definování kamery, tedy bodu, ze kterého bude scéna vykreslována. Dále následuje nastavení osvětlení, objektů, materiálů aj. komponent 3D scény.
2. **Vertex shader:** Program, jehož hlavním úkolem je transformovat 3D souřadnice vrcholu z objektového prostoru do prostoru kamery (tedy relativizovat souřadnice vůči pohledové kameře) a aplikovat požadovanou projekci. Tyto transformace obvykle zahrnují operace jako posunutí, rotaci a změnu velikosti definované jedinou transformační maticí. Dále může *vertex shader* provádět výpočty osvětlení, mapování textur a další operace na úrovni vrcholu, aby připravily vrchol pro vykreslování.
3. **Určení viditelných částí (culling):** Ořezávání je proces, při kterém jsou odstraněny části 3D objektů, které nejsou viditelné v zorném poli kamery (*frustum*) popř. části objektů, které jsou zakryté jinými objekty. Ořezávání pomáhá ušetřit výpočetní výkon tím, že se neprovádí vykreslování neviditelných částí.
4. **Rasterizace:** Proces kdy se body na povrchu 3D modelu převádějí na pixely na obrazovce s cílem vytvořit "fragmenty". Při rasterizaci se určuje, které body nebo vrcholy 3D objektu spadají do pixelového rastru na obrazovce. Atributy vertexů v rámci jednoho pixelu jsou interpolovány. Interpolace probíhá zpravidla na attribitech polohy, normálových vektorů, barev aj.

5. **Fragment Shader:** Když je pixel vygenerován během rasterizace (převodu vrcholů na pixely), fragment shader bere do úvahy různé informace, jako jsou barvy textur, umístění světla a pozici kamery, a na základě těchto informací vypočítá konečnou barvu pixelu.
6. **Vykreslení (render):** Závěrečný krok v procesu zpracování grafiky, při kterém se barvy a vlastnosti aplikují na pixely na obrazovce. Zahrnuje některou logiku, jako je míchání (*blending*) a test hloubky (*depth testing*).



Obr. 12 *Graphics rendering pipeline*. vlastní tvorba dle (Ghayour, Cantor 2018; Ariën 2017; Sherif 2018).

### 3.6.2 Komponenty 3D vizualizace

Systémy umožňující 3D vizualizaci popř. virtuální realitu se ve většině případů skládají z obdobných komponentů. Následující kapitola zahrnuje obecný popis klíčových komponent pro tvorbu 3D scény.

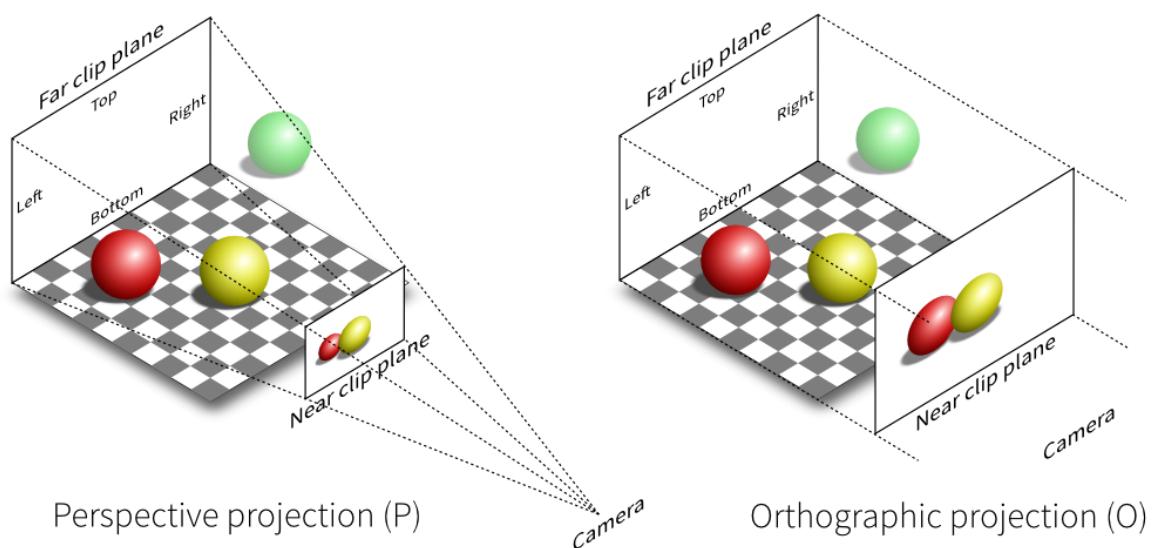
**Scéna:** Scéna je jedním z klíčových prvků virtuální reality a 3D vizualizací. Představuje digitální prostředí, ve kterém se odehrává uživatelský zážitek. Scéna funguje jako základní stavební blok, do kterého jsou umisťovány všechny objekty. Struktura scény je zpravidla implementována pomocí hierarchické stromové datové struktury obdobně jako je DOM v prohlížečích. Tato struktura se nazývá graf scény (*scene graph*). Scéna zpravidla obsahuje druhy objektů, které jsou Světla, Objekty, Meshy. Mesh tedy objekt, který se skládá z Geometrie (tvaru) a Materiálu (pravidel, jak má mesh vypadat). Geometrie je datová struktura, která reprezentuje polohová data jednotlivých vertexů. Zdrojem geometrie je zpravidla soubor v daném formátu (např. gltf, obj. aj.).

**Mesh** je tedy sémantická struktura pro tvorbu objektů z geometrie a jednoho či více materiálů. Nejběžnější geometrickou reprezentací je indexovaná polygonová síť (mesh). Hardware tedy GPU jsou optimalizovány pro práci se sítí trojúhelníků (*triangle mesh*) a to z toho důvodu, že jakýkoliv polygon je možné rozdělit na trojúhelníky. Objekt je definován trojúhelníky, které se skládají z vertexů, hran a ploch. Objekty jsou reprezentovány jako pole souřadnic vertexů a pole trojúhelníků, které tvoří. (Dunn, Parberry 2011)

**Kamera:** Kamera určuje pohled uživatele do virtuálního světa. Hlavní dělení typů kamer je dle typu zobrazení na **perspektivní** a **ortografické**. **Perspektivní kamera** vytváří scénu s iluzí hloubky. Objekty vzdálenější od kamery jsou zobrazovány menší než objekty blíže ke

kameře, což vytváří dojem trojrozměrného prostoru. **Ortografická kamera** zobrazuje objekty ve scéně bez perspektivy. To znamená, že objekty mají stálou velikost, bez ohledu na jejich vzdálenost od kamery. Tento typ kamery se často používá pro technické výkresy, architektonické plány a 2D vizualizace, kde je důležitá přesná geometrie. Klíčové parametry virtuální kamery jsou (Dunn, Parberry 2011; three.js Contributors 2023a):

- **Zorný úhel:** Zorný úhel (*FOV - field of view*) určuje, jak široký úhel zobrazení kamera má.
- **Poměr stran (aspect ratio):** Poměr stran definuje proporce mezi šírkou a výškou obrazovky nebo zobrazovacího okna kamery. Správný poměr stran je důležitý pro zachování proporcí objektů ve scéně.
- **Výřezové roviny (clipping planes):** určují, které objekty jsou viditelné a které ne. Kamera může mít dvě výřezové roviny: blízkou (*near*) a vzdálenou (*far*). Objekty mimo tyto roviny nejsou vykresleny (viz. *frustum culling*).



Obr. 13 Příklady perspektivní a ortografické kamery. (three.js Contributors 2023a)

**Materiály:** „Parametrický popis vizuálních vlastností reálného objektu reprezentovaného síťovým primitivem (mesh).“ (Khronos Group 2021) Materiály určují vzhled a povrchy objektů ve scéně. Jedná se o soubory vlastností a charakteristik, které definují, jak objekt interaguje se světem a jak se jeví pozorovateli. Zahrnují textury, barvy, lesk a další optické vlastnosti, které dávají objektům jejich vizuální charakter (Blender Documentation Team 2023a). Způsob, jakým jsou tyto charakteristiky zpracovány se pak nazývá *shading model*. Mezi tyto modely se řadí **Lambertův**, **Phongův**, **Blinn-Phongův** a **PBR (Physically Based Rendering)**. Jednotlivé modely umožňují simulaci způsobů, jakým světlo interaguje s různými typy povrchů. **Lambertův model** – simuluje jakým způsobem světlo interaguje s hrubými matnými povrhy, **Phongův** – jakým způsobem se světlo odráží od lesklých povrhů a **PBR** model simuluje fyzikální chování světla. Tato simulace je provedena na základě parametrů: Base Color (základní barva), Roughness (hrubost), Metalness (do jaké míry se povrch chová jako kov), Specular (Odrazivost). Zároveň PBR model je schopný simulovat, jakým způsobem se světlo chová v průsvitných, popř. částečně průsvitných materiálech. (Chow 2018; Dunn, Parberry 2011) PBR model je standardem pro simulaci interakce světla s objekty napříč vykreslovacím softwarem. Jelikož je PBR založen na simulaci chování světla na základě

reálných fyzikálních procesů, je nutné při tvorbě VP brát v potaz fyzikálně přesné velikosti scén. Je nutné tedy adaptovat velikost VP na jednotky intenzity světel. Tedy při simulaci žárovky o 100 wattech není možné osvětlovat prostor větší než standardní místnost (Discover three.js Contributors 2023).

**Textury:** Textury jsou grafické vzory, které se aplikují na povrchy trojrozměrných objektů. Tyto vzory mohou obsahovat detaily a barvy. Jsou klíčovým prvkem pro definici vizuálního vzhledu a povrchového charakteru objektů. V kontextu materiálů jsou textury využity k definici vlastností, jako jsou barva, lesk, hrubost a další optické charakteristiky. Textura je obyčejná 2D bitmapa, kterou je nutné namapovat na 3D objekt, skrze proces UV mapování, tedy přiřazení souřadnic na povrchu objektu souřadnicím v rámci textury (Dunn, Parberry 2011). Texturování lze použít i pro definování jiných charakteristik než barvy povrchu. Běžnou technikou je tzv. *bump mapping / normal mapping*, kdy informace uložená v textuře modifikuje normálový vektor povrchu na úrovni jednotlivých pixelů, což při osvětlení modelu umožňuje vytvořit iluzi podrobnější geometrie (Dunn, Parberry 2011). Pro práci s texturami ve webovém prostředí je důležité brát v potaz jejich velikost a detail za účelem minimalizace času stažení textury a množství operační paměti, kterou zobrazení textury vyžaduje (Hutter 2021; three.js Contributors 2023e). Textury lze ve vykreslovacích engineech využít k projektování videa popř. jiných vstupů (např. <canvas> html element) (Discover three.js Contributors 2023).

**Osvětlení:** Osvětlení v reálném světě vzniká pomocí nekonečného množství paprsků odrážejících se nekonečné mnohokrát od prostředí se slábnoucí energií, dokud netrefí oko, popř. kameru. Takováto simulace v počítači není možná, tudíž vznikly abstrakce přímého, nepřímého osvětlení (Dunn, Parberry 2011).

- Přímé (Direct) – Světlo dopadá ze zdroje přímo na objekt.
- Nepřímého (Indirect) – Světlo odražené od jiných objektů ve scéně.

Obecně se světla ve vykreslovacích engineech dělí na (Unity 2022; three.js Contributors 2023b; Blender Documentation Team 2023b; Dunn, Parberry 2011):

- **Point Light** – Jedná se o bod, ze kterého je vyzařováno světlo do všech stran. Síla světla zpravidla slabne se vzdáleností od něj.
- **Spot Light** – Obdobné jako Point Light, světlo je ve tvaru kuželeta, který se rozšiřuje se vzdáleností od zdroje světla.
- **Area Light** – Světlo, jehož zdrojem je plocha, popř. povrch.
- **Sun Light / Directional Light** – Světlo, které vychází z nekonečně vzdáleného bodu v daném směru.
- **Ambient Light** – Světlo osvětlující veškeré objekty stejně, ze všech směrů.

**Stíny:** Simulace stínu je možné dosáhnout pomocí techniky stínového mapování (*shadow maps*), kdy pro každý zdroj světa, jsou všechny objekty vykreslovány z pohledu daného světla. Tento způsob vede k opakování vykreslení celé scény (*draw call*) pro každé světlo plus pro samotnou kameru (three.js Contributors 2023d). Simulace stínů v 3D aplikaci je úzce spjatá s optimalizací výkonu, jelikož se jedná o náročnou operaci. Hlavní dva přístupy simulace stínů je dynamický (*shadows maps*) a statické mapování, kdy simulované stíny jsou integrovány do textury (*baked shadows*)<sup>2</sup>.

---

<sup>2</sup> Termíny stínové a světlené mapování jsou zaměnitelné.

### 3.6.3 Vývoj

Vývoj aplikací virtuální reality má řadu omezujících požadavků, a to především ze strany optimalizace výkonu, interakce a specifických pracovních postupů (iterace testování aplikací napříč zařízeními). V případě vývoje pro web přibývají další požadavky ve formě optimalizace doby načítání (přenos dat po síti), limitace grafických API, nutnost vysoce optimalizovaných 3D modelů aj. (Oculus VR 2022) Tyto specifika je při vývoji nutné mít na paměti a upravit dle nich návrhová rozhodnutí a tedy i výběr technologie.

### Architektura

Vývoj virtuálních zážitků sdílí velkou část problematiky s herním vývojářstvím. Jedná se o tvorbu virtuálního světa, ve kterém je jeden nebo více uživatelů, kteří se světem interagují, a to skrze pohyb ve světe či s jeho objekty. Z tohoto důvodu vývoj virtuálních zážitků a tvorba her sdílí i některé návrhové vzory. Jedním z těchto vzorů je ECS (*Entity component system*). ECS představuje způsob, jak abstrahovat a strukturovat systémy rozdělní na systémy, entity a komponenty. Komponenty jsou soubory specifických typů dat (udržují stav), entity jsou pak indexované soubory jednoho či více komponentů. Veškerá funkcionality je následně definována v rámci systémů, které přistupují ke komponentům a mění jejich stav. Za základní pravidla ECS je možné považovat výroky: komponenty nemají funkcionality, systémy neudržují stav (data), entity samotné jsou indexy (Rez Bot 2018). Tím se vytváří modulární a flexibilní architektura, která umožňuje kompozici různých kombinací dat a chování a dále usnadňuje tvorbu, úpravu a rozšiřování virtuálního světa. V praxi pak entitu může být např. miniatuра 3D modelu města, které je přiřazen komponenty, které umožňují rotaci pomocí kurzoru aj. Tento přístup využívá řada herních enginů (Unreal Engine, Unity, Godot, Wonderland engine) a také webové řešení jako Babylon.js a A-Frame popř. řešení jako Mozilla Hubs. (Stapley 2022; Ford 2017; Mozilla Hubs 2022)

### 3.6.4 Výkon

Při vývoji VR aplikace pro web. Je nutné mít na paměti parametry ovlivňující virtuální zázitek. Jedná se primárně o přenosovou náročnost tedy velikost dané scény (bajty) a následně výpočetní, popř. vykreslovací náročnost. Velikost scény přímo ovlivňuje čas, jaký je potřeba pro její stažení. V případě webového prostředí nelze počítat s tím, že každé zařízení má rychlé připojení k internetu. Výkon se většinou měří pomocí velikosti využité RAM, počtu vykreslovacích příkazů (*draw calls*), a snímková frekvence aplikace; FPS (*frames per second*), které je aplikace schopná vykreslit. Z hlediska výkonu hrají roli při optimalizaci výkonu pak (Mozilla Corporation 2023b):

- **Počet polygonů** – Počet polygonů by zpravidla neměl zasahovat do řádu statisíců.
- **Počet materiálů** – Vyšší počty materiálů zvyšují počet potřebných vykreslovacích příkazů, tedy zatěžují GPU zařízení.
- **Velikost a počet textur** – Textury je zpravidla nutné nejvíce optimalizovat, a to jak jejich velikost, tak jejich rozlišení. Ve vykreslovacím enginu three.js textura zabere  $\text{šířka} * \text{výška} * 4 * 1.33$  bajtů operační paměti. Což pro texturu o rozlišení 1024x1024 znamená 5.19 MB operační paměti. (three.js Contributors 2023e) Optimalizace je často řešena kompresí.
- **Počet světel** – Větší počet dynamických světel znamená větší počet vykreslení a větší výpočetní náročnost pro simulaci osvětlení.

## 4 ANALÝZA TECHNOLOGIÍ

Stěžejní kapitola práce se zaměřuje převážně na analýzu technologií, které mohou tvořit vhodný tzv. *Virtual World Generator* (viz. kap. 3.2 Systém virtuální reality) pro účely vizualizace geografických dat na webu. Za účelem úspěšné analýzy je vhodné dostupné technologie klasifikovat. Klasifikačními kritérii v tomto případě je obor a primární účel v jakém technologie figuruje. Takto je možné definovat kategorie na CAD, GIS, herní vývojářství, 3D modelování (umění) aj. Dalším klasifikačním kritériem je následně zda se jedná o technologie nativní či webové. Z hlediska vývoje je vhodné klasifikovat technologie na základě míry abstrakce jakou uživateli poskytují, tedy zda se jedná o psaní kódu pro grafické shadery či o knihovnu až po kompletně řešenou aplikaci s GUI. Proces vývoje aplikace přesahuje přes více zmíněných účelů užití, napříč mírami abstrakce a využívá nativních i webových řešení. Analýz dostupných technologií tedy vyžaduje přehled technologií napříč zmíněnými kategoriemi, které umožní pořízení, úpravu a vizualizaci dat, interakci a následně publikaci. Nejedná se tedy o jednotlivou technologie ale jejich soubor (*tech stack*). Na základě této skutečnosti je tedy nutné hodnotit i vzájemnou kompatibilitu jednotlivých technologií, což může přinést výraznou míru komplexity, jelikož je nutné technologie na různých úrovních kombinovat. Za účelem získání reprezentativních výsledků je nutné hodnotit vhodnost jednotlivých technologií v *tech stacku* v kontextu specifického využití.

Obecný postup tvorby virtuálního prostředí lze rozdělit do několika kroků. Prvním úkolem je získání dat pro zobrazení, což se dosahuje pomocí technologií jako je Lidar, fotogrammetrie a tradiční mapování pro získání potřebných dat. Následuje fáze zpracování dat, kde se vytváří 3D objekty (mesh), zpracovávají textury a vytváří celková 3D scéna na základě nasbíraných informací. Dalším krokem je vizualizace dat, kde jsou tyto informace prezentovány pomocí vybraného vykreslovacího enginu a implementací aplikační logiky. Nakonec dochází k publikaci dat, kdy je vytvořená vizualizace zveřejněna na internetu, podstoupí testování a optimalizaci pro dosažení optimálního výkonu a širší dostupnosti.

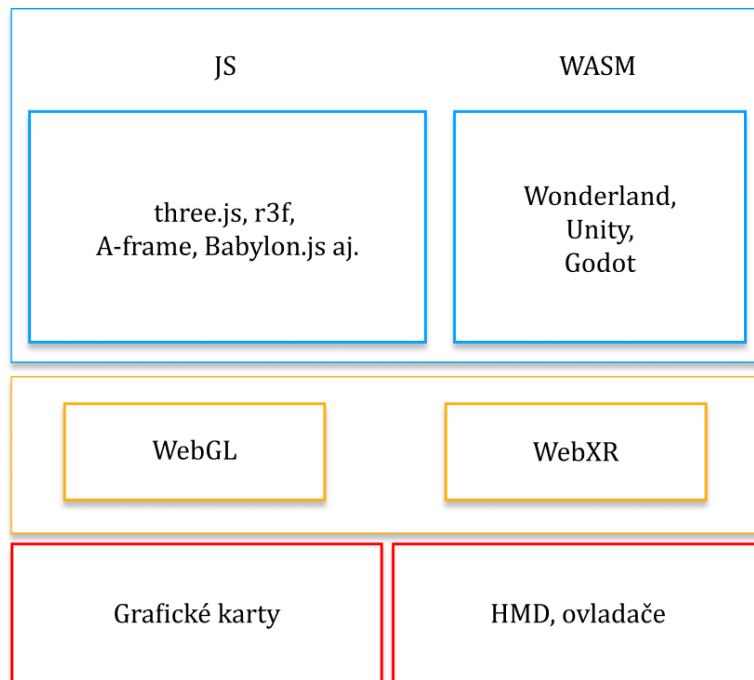
### 4.1 Taxonomie technologií

Z hlediska funkce kategorizují (Stachon, Kubicek, Herman 2020) do třech hlavních skupin na CAD (computer aided design), GIS, programy pro tvorbu 3D grafiky a fotogrammetrické programy. Dále zmiňují i herní enginy a webové technologie. Autoři dále zmiňují dělení na webové aplikace a nativní software. Za nativní software – vyvinutý pro daný operační systém a jehož runtime prostředí je specifické pro daný hardware a operační systém – je možné považovat GIS a CAD řešení (QGIS, ArcGIS, GRASS, FME, Autodesk, Bentley Microstation aj.), aplikace pro tvorbu 3D grafiky (Blender, SketchUP, Microstation, Rhinoceros 3D aj.) a herní enginy (Unity, Unreal Engine, Godot, Wonderland). Webový software, tedy software, jehož runtime prostředí je buďto v rámci klienta, tím pádem v prohlížeči uživatele nebo v rámci serveru. Webové technologie dříve závislé na externích rozšířeních do vybraných internetových prohlížečů, jsou nyní nahrazeny standardizovanými technologiemi jako je HTML5, WebGL, WebXR popř. rozsáhlý ekosystém knihoven a frameworků v jazyce JavaScript usnadňující tvorbu virtuálních prostředí (Stachon, Kubicek, Herman 2020).

Pro dosažení daného případu užití je v mnoha případech využíváno technologií napříč všemi zmíněnými kategoriemi. Zde je nutné zmínit, že pro vývoj výsledné webové vizualizace neznamená využití pouze webových technologií. Tento fakt je prominentní především při přípravě a zpracování geoprostorových dat. Za účelem výběru technologie pro vývoj VR aplikace na webu zmiňuje (Godber 2022) 4 základní postupy:

- Přímý vývoj nad WebGL a WebXR
- Vývoj skrze Javascriptový framework / knihovnu pro renderování a pro WebXR.
- Vývoj v rámci desktopového herního enginu a export WebGL aplikace skrze WASM (*Web Assembly*).
- Vývoj skrze dedikovaný WebXR engine.

Pro účely geoprostorové vizualizace je možné vyloučit postup č. 1, jelikož se jedná o příliš nízko-úrovňový vývoj. V rámci této práce však je role základních API představena (viz. kap. **Error! Reference source not found. Error! Reference source not found.**). Hlavním zaměřením analýzy technologií je postup č. 2 a č. 4, kdy vývoj probíhá na základě standardních webových technologií. Postup č. 3 je méně flexibilní, jelikož exportéry virtuálních prostředí vytvořených v herních enginech skrze WASM jsou často černými skříňkami. Alternativně, existují nástroje, které se snaží tento problém řešit propojením postupu 2 a 3. Postup č. 4 je uživatelsky nejpřívětivější cestou, jelikož umožňuje tvorbu VP v rámci GUI a přímou kompatibilitu s WebXR API.



Obr. 14 Taxonomie webových technologií umožňujících tvorbu virtuálních prostředí dle míry abstrakce.  
 Červeně – hardware, žlutě – prohlížeč, modře – JS ekosystém  
 upraveno dle: (Godber 2022)

## 4.2 Webový vývoj

Volba webového prostředí přináší jisté benefity, ale i překážky při tvorbě VR aplikací. Primárním benefitem oproti desktopovým aplikacím je dostupnost (*availability*). Jelikož web je platformě a hardwarově agnostický, tedy je možné k němu přistupovat takřka skrze veškerý běžně užívaný hardware a software. V případě 3D vizualizace a virtuální reality je dostupnost a přístupnost v posledních letech umožněna díky výraznému vývoji HTML5, WebGL a WebXR. Dalším z benefitů je fakt, že většina globálně užívaných aplikací se postupně přesouvá z desktopových řešení do webového prohlížeče (kancelářské programy: Microsoft Office Suite, grafické: Figma, Canva, vývojářské: CodePen, VSCode web aj.) (Řeháček 2020). Tvorba klasických webových aplikací je umožněna pomocí kombinace technologií tzv. *web-standard technologies* (Řeháček 2020) jimiž jsou:

1. HTML – značkovací jazyk určující strukturu a obsah webové stránky
2. CSS – kaskádový stylovací jazyk určující vzhled obsahu stránky
3. SVG – značkovací jazyk umožňující 2D vektorovou grafiku na webu
4. JavaScript – vysokoúrovňový jazyk umožňující interakci s obsahem a vzhledem

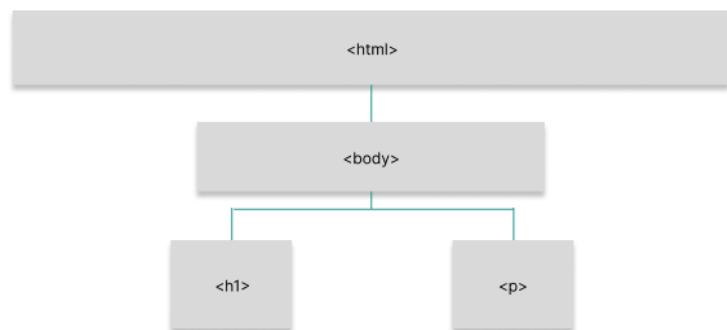
Mezi webové standardy je pak volněji možné zařadit i klíčové API (DOM, Fetch API, WebGL, WebXR aj.), které umožňují snazší vývoj pro webové prostředí.

#### 4.2.1 Web API

##### DOM API

Při zpracování HTML dokumentu prohlížečem vzniká abstraktní stromová datová struktura, která slouží k vykreslení webové stránky. Tato struktura je následně přístupná skrze rozhraní nazývané *Document Object Model* (DOM) (Řeháček 2020). DOM reprezentuje dany HTML dokument tak, že umožňuje manipulaci s jeho strukturou, stylem a obsahem pomocí skriptovacího jazyka, například JavaScriptu (MDN Contributors 2022a). Z hlediska HTML je to objektový model, který definuje atributy, metody a události. Z hlediska JavaScriptu jde o API, které umožňuje interakci s HTML dokumentem.

```
<!DOCTYPE html>
<html>
  <body>
    <h1>Titulek</h1>
    <p>Odstavec</p>
  </body>
</html>
```



Obr. 15 Zdrojový kód HTML a DOM struktura. upraveno podle: (Řeháček 2020; W3Schools 2023)

##### WebGL API

Jedná se o nízko úrovňové Javascriptové aplikaci rozhraní, které umožňuje vysoko výkonné vykreslování interaktivní 3D a 2D grafiky v rámci kompatibilního webového prohlížeče. Díky tomu, že WebGL následuje implementaci OpenGL ES (*OpenGL for Embedded Systems*<sup>3</sup>) je možné v rámci webových stránek využívat výpočetní síly grafických karet v zařízení uživatele (MDN Contributors 2022b). WebGL je přístupné skrze HTML5 Canvas element. Důležité je, že je to nativní součást prohlížečů, a proto nevyžaduje žádné další doplňky (Khronos Group 2023). Pro vykreslování využívá WebGL jazyk GLSL, který popisuje způsob, jakým je daný objekt vyobrazen (*shader*). Základní funkčnost WebGL dále rozšiřuje Javascriptové knihovny, jež zjednodušují složitý proces grafického vykreslování přes jednodušší API. Od roku 2011, kdy byl WebGL poprvé zaveden, umožnil přímý přístup k GPU prostřednictvím JavaScriptu, a v roce 2013 byla představena verze WebGL 2. Dodnes zůstává standardním prostředkem pro zobrazování složitější grafiky na webu. Jednoduše řečeno, WebGL je soubor javascriptových funkcí umožňující webovým prohlížečům zobrazovat 3D grafiku s využitím grafických karet.

<sup>3</sup> WebGL je založen na specifikaci OpenGL ES 2.0 s cílem maximalizovat přenositelnost na mobilní zařízení namísto OpenGL, který je pro desktopová řešení.

## WebXR API

WebXR je specifikace definovaná v rámci W3C skupinou pro imerzní web, za účelem poskytnutí jednotné komunikace mezi VR a AR hardwarem a webovým prostředím. WebXR API umožňuje jednotný přístup k vytváření virtuálních zážitků pro mobilní i desktopový VR hardware. WebXR API je založeno na OpenXR<sup>4</sup> specifikaci skupiny Khronos. Hlavními body zájmu WebXR API je detekce a vyhledání možností daného HW a následně správné zobrazení (s odpovídající snímkovou frekvencí) obsahu v rámci HW (Immersive Web Working Group 2023; tro to WebXR and A-Frame Part 1 2021). WebXR podpora v rámci prohlížečů a jednotlivých VR zobrazovacích zařízeních je klíčová v případě hodnocení přístupnosti. WebXR operuje na základě relací (*session*). Knihovna umožňuje tři typy relací a to (MDN Contributors 2023b):

- ***immersive-vr***: Umožňuje přístup k imerzním u zobrazovacímu zařízení a umožňuje zobrazení virtuálního obsahu v plném VR režimu.
- ***immersive-ar***: Umožňuje přístup k imerznímu zařízení, s tím že vykreslený obsah je kombinován s reálným světem. Způsob prolnutí reálného světa (vstupu z kamery) se řídí pomocí *environmentBlendMode* parametru.
- ***inline***: Umožňuje zobrazit virtuální obsah v rámci standardního HTML dokumentu, aniž by vyplnil celou obrazovku. Umožňuje více relací na jedné stránce a nevyžaduje speciální hardware. V případě systémů s IMU umožňuje rotační snímání.

Aplikace založená na WebXR API musí následovat tzv. životní cyklus VR webové aplikace, který spočívá v (Immersive Web Working Group 2022; MDN Contributors 2023b):

1. Dotaz na to, zdali je XR podporováno (prohlížečem / zařízením) a jaký typ relace je podporován, popř. vyžadován.
2. Pokud je podpora k dispozici, informovat uživatele o dostupnosti vyžadované XR funkcionality např. v podobě přidání tlačítka umožňujícího spuštění virtuálního prostředí.
3. Uživatelská událost (*event*) signalizující, že uživatel má zájem o spuštění XR relace.
4. Požadavek na relaci, kde je možné specifikovat typ (*inline*, *immerse-vr*, *immerse-ar*) a vlastnosti (např. možnosti pohybu uživatele poskytnuté daným zařízením, které by požadovaná relace měla splňovat).
5. Vytvoření vykreslovacího cyklu, který obnovuje senzorická data (poloha, pohyb atd.) a který generuje snímky na obrazovku zařízení.
6. Vykreslování, dokud uživatel nespustí událost ukončení XR relace.
7. Ukončení XR relace.

Samotné WebXR API zprostředkovává pouze jednotný interface pro přístup k funkcionalitě vstupních a výstupních zařízení. Za účelem vytvoření virtuálního zážitku je tedy nutné propojení s animačním cyklem definovaným v rámci některé z technologií umožňující renderování 3D grafiky na webu (WebGL + JS knihovny) skrze XRWebGLLayer.

## WebGPU API

WebGPU je dalším krokem v evoluci webových grafických API, tedy zamýšleným následníkem WebGL. Jedná se o experimentální aplikační rozhraní, které umožňuje vysoko výkonné 3D

---

<sup>4</sup> OpenXR je specifikace pro standardizaci rozhraní pro vývoj aplikací pro virtuální a rozšířenou realitu. Jejím cílem je umožnit interoperabilitu mezi různými platformami a technologiemi pro webovou XR.

a 2D vykreslování na webu. WebGPU je nízko úrovňové API, které poskytuje programátorům přímý přístup k hardwarové akceleraci na grafických kartách. Rozdílné od WebGL, které pro přístup k GPU využívá OpenGL ES APIs (Introducing WebGPU 2023). Jedná se o technologii ve vývoji, tudíž podpora mezi prohlížeči je velice omezená. Momentálně je WebGPU podporováno jen v nových verzích Chrome, Edge and Opera (Can I Use 2023b).

#### 4.2.2 Prohlížeče

Webové prostředí je široký a různorodý ekosystém technologií. Za účelem vývoje úspěšné aplikace je nutné zohlednit aspekt kompatibility dané aplikace s webovým prohlížečem (*runtime aplikace*). Kompatibilita je zajištěna tak, že verze daného prohlížeče podporuje jazyky, knihovny a frameworky použité při vývoji aplikace. V případě virtuální reality je klíčová podpora WebGL API a WebXR API. Za účelem porovnání je vhodné mít přehled o aktuálních webových prohlížečích.

Prohlížeče je možné rozdělit podle platformy pro kterou jsou implementovány. Tradiční dělení je na (desktop, mobilní) v případě VR je nutné brát v potaz i prohlížeče vyvinuté speciálně pro HMD (*Wolvic, Meta Oculus Browser*). Analýza podpory prohlížečů pro WebXR API viz. Tab. 4. V rámci tabulky jsou zahrnutы prohlížeče, které implementují WebXR API. Pro podporu WebGL není nutná podrobná analýza, jelikož všechny zmíněné prohlížeče v Tab. 4 plně podporují WebGL 1.0 a WebGL 2.0 (Can I Use 2023a).

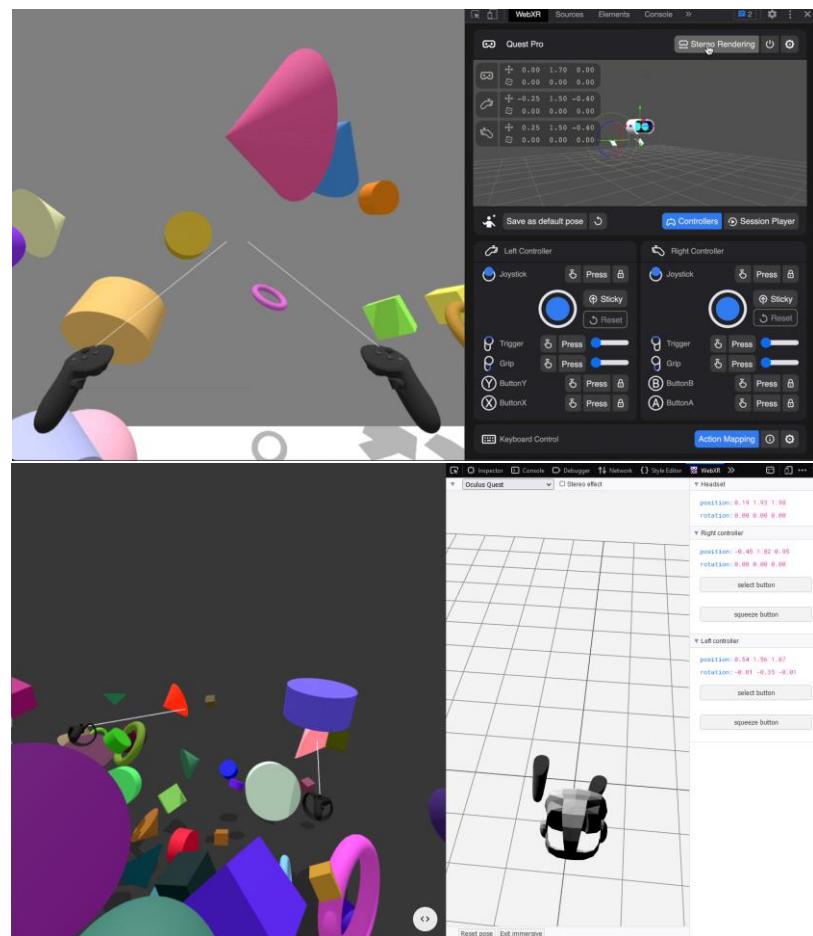
**Tab. 4 Podpora WebXR Device API ve verzích vybraných prohlížečů.** – <sup>1</sup>WebXR Device API je rozsáhlé API, které je stále ve vývoji, tudíž není možné přesně určit míru podpory, <sup>2</sup> Prohlížeče API nepodporují defaultně, <sup>3</sup> Globální zastoupení prohlížečů na trhu - (k datu: 28.1.2023), sestaveno podle: (StatCounter 2023; Meta 2023; W3C 2023; Can I Use 2023c; Igalia SL 2023)

Prohlížeče		Částečně <sup>1</sup>	Nutné povolit <sup>2</sup>	Zastoupení [%] <sup>3</sup>
Desktop	Chrome	79.0 - 112.0	-	22.71
	Edge	79.0 - 109.0	-	4.17
	Safari	-	13.0 - 16.2	3.12
	Firefox	-	77.0 - 111.0	3.05
	Opera	66.0 - 92.0	52.0 - 65.0	0.70
	IE	-	-	0.73
Mobilní	Chrome - Android	109.00	-	41.54
	Samsung Internet	12.0 - 19.0	-	3.20
	Opera Mobile	72.00	-	0.01
	Firefox for Android	-	107.00	0.29
	Safari	-	-	14.67
	WebXR Viewer (IOS)	1.0 - 2.0	-	-
HMD	Wolvic	0.9.5 - 1.2	-	-
	Meta Quest Browser	5.0 - 16.2	-	-

#### 4.2.3 Vývojářské nástroje

Vývoj virtuálních imerzních prostředí z velké části na tradiční obrazovce vyžaduje dodatečné nástroje umožňující prototypování a testování při vývoji. Dostupnost HMD zařízení stále není na takové úrovni, aby bylo možné předpokládat, že uživatel bude virtuální prostředí zažívat skrze HMD. Tuto problematiku je možné řešit v rámci návrhu aplikace. Popř. druhou možností částečného řešení dostupnosti aplikace jsou existující nástroje jako: **Immersive web emmulator** (Chrome, Edge) a **WebXR API emulator** (Firefox) (Meta Quest 2023). Rozšíření do prohlížečů, které umožňují uživateli simulovat interakci aplikace s HMD zařízením. Další

přístupy k prototypování je přímo vývoj ve webovém prostředí. Tento způsob vývoje umožňuje platforma Glitch.com, která poskytuje textový editor, vývojový server a také způsob publikování. V případě prototypování v tradičním mobilním zařízení je možné propojit mobilní zařízení s počítačem skrze USB kabel a pomocí vývojářských nástrojů prohlížeče Chrome streamovat lokální vývojový server do mobilního zařízení (Basques 2023). Druhou možností, jak testovat při vývoji, je skrze nastavení CI / CD procesu<sup>5</sup>. V rámci platformou Github a Gitlab lze využít služby, kdy při každé změně v zdrojovém kódu v rámci sledovaného reposítáře je webová aplikace vystavěna a publikována. Tento přístup zároveň umožňuje univerzální testování, jelikož aplikace je přístupná všem zařízením, které mají webový prohlížeč a přístup k internetu. Tento postup byl zvolen i při průzkumu technologií a následném vývoji výsledné aplikace (viz. kap. 5.3.3 CI / CD).



**Obr. 16 Emulátory HMD zařízení v prohlížeči.** nahoře - Immersive web emmulator, dole - WebXR API emmulator.

### 4.3 Analýza technologií

Technologie umožňující 3D vizualizaci na webu je možné obecně popsat jako abstrakce nad WebGL API. Následující kapitola je výsledkem analýzy a praktického porovnání technologií na základě vytvoření jednoduché scény. Primárním hodnotícím kritériem technologií je jejich kompatibilita, popř. podpora (existující nástroje) pro WebXR API, jehož implementace je

<sup>5</sup> CI/CD proces (*Continuous Integration / Continuous Deployment*) je postup automatizovaného testování a nasazování softwarového produktu v průběhu jeho vývoje. Tímto způsobem je zajištěno pravidelné integrování a nasazování nových změn do aplikace, což usnadňuje rychlý vývoj a zajišťuje konzistentní kvalitu (RedHat 2022).

klíčová pro existenci virtuální reality na webu. Výběr technologií pro analýzu a popis byl založen jednak na rešerší literatury dokumentující existující řešení, seznamu technologií (Seguin 2023), tak na základě konzultace s komunitou uživatelů a vývojářů jednotlivých technologií.

#### 4.3.1 Geoprostorová řešení

Jelikož je práce zaměřena na geoprostorová data, je logické prozkoumat možnosti tvorby virtuálních prostředí, které poskytované běžnými GIS. Primárním zaměřením GIS je především manipulace, analýza a následně 2D neimerzní neinteraktivní vizualizace dat. Je však možné nalézt implementace, popř. rozšíření / kombinace s jinými technologiemi, které umožňují vizualizaci dat ve virtuálních prostředích.

##### Proprietární řešení

Z proprietárních řešení je vhodné zmínit ESRI řešení, jakožto prominentního dodavatele GIS softwaru. V rámci desktopového softwaru ArcGIS Pro je možné dosáhnout úrovně Desktop VR, tedy interaktivní zobrazení a analýzy nad 3D modely. Možným řešením pro tvorbu urbánních virtuálních prostředí je software **Esri City Engine**, který je primárně určen pro modelování a vizualizaci městské infrastruktury a zástavby. V případě propojení s externím softwarem poskytuje ESRI možnost propojení s herními enginy Unity a Unreal Engine skrze poskytované SKD (*Software Development Kit*) (ESRI 2023c). V případě virtuální reality na webu pak umožňuje ESRI publikaci 3D scén z ArcGIS Pro na ESRI cloud řešení ArcGIS Online, které umožňuje prohlížení daných scén, stále ale na úrovni neimerzní VR. Na základě takto publikovaných scén je následně možné vyvíjet webové aplikace s přidanou funkcionality pomocí **ArcGIS JS API** (ESRI 2023a; KAM Brno 2023). Immersive VR úrovně lze dosáhnout využitím City Engine, který umožňuje export sady 360° snímků z daného 3D modelu na ArcGIS Online a prohlížení těchto snímků z prostředí prohlížeče a stereoskopického HMD zařízení. (ESRI 2023b; Plačková 2022) Tato funkcionality však neumožňuje pohyb scénou, jelikož se jedná pouze o snímky.

Mezi proprietární řešení je následně možné zařadit i řešení společnosti Hexagon a to primárně **LuciadRia JS API**, což je univerzální API pro vizualizaci dat v prostředí prohlížeče pomocí WebGL (Hexagon 2023). Stejně jako u ESRI řešení neposkytuje Hexagon imerzní VR funkcionality. V rámci této práce nejsou proprietární řešení primárním zájmem.

##### Open Source řešení

Mezi Open Source GIS řešení je nejvíce prominentní QGIS. Podpora pro práci s 3D daty je v QGIS základní, Samotný QGIS přímo nepodporuje publikaci do webového prostředí jako ArcGIS. Této funkcionality lze však dosáhnout pomocí zásuvného modulu **qgistothree.js**, který jak jméno napovídá umožňuje vytvořit kompletní three.js scénu publikovatelnou jakožto webovou stránku přímo z QGIS GUI. Jedná se o velice hodnotný nástroj pro konvertování GIS dat do webového prostředí. Samotnou knihovnu three.js není možné považovat za GIS řešení, jelikož je to obecná knihovna, která umožňuje tvorbu 3D prostředí skrze JavaScript a nativně nepodporuje geoprostorovou lokalizaci. Zároveň je nutné zmínit, že takto publikované scény jsou na úrovni imerze Desktop VR, každopádně three.js podporuje integraci s WebXR API, tudíž teoreticky umožňuje dosažení imerzní VR úrovně.

Při přesunu od desktopových řešení k čistě webovým, je možné zmínit **CesiumJS**, **MapboxGL**, **deck.gl**, **vts-geospatial**, **ITowns**. Porovnání těchto knihoven / frameworků provedli (Horký 2020; 2019 - Battle of 3D Rendering Stacks 2019; Peters et al. 2021).

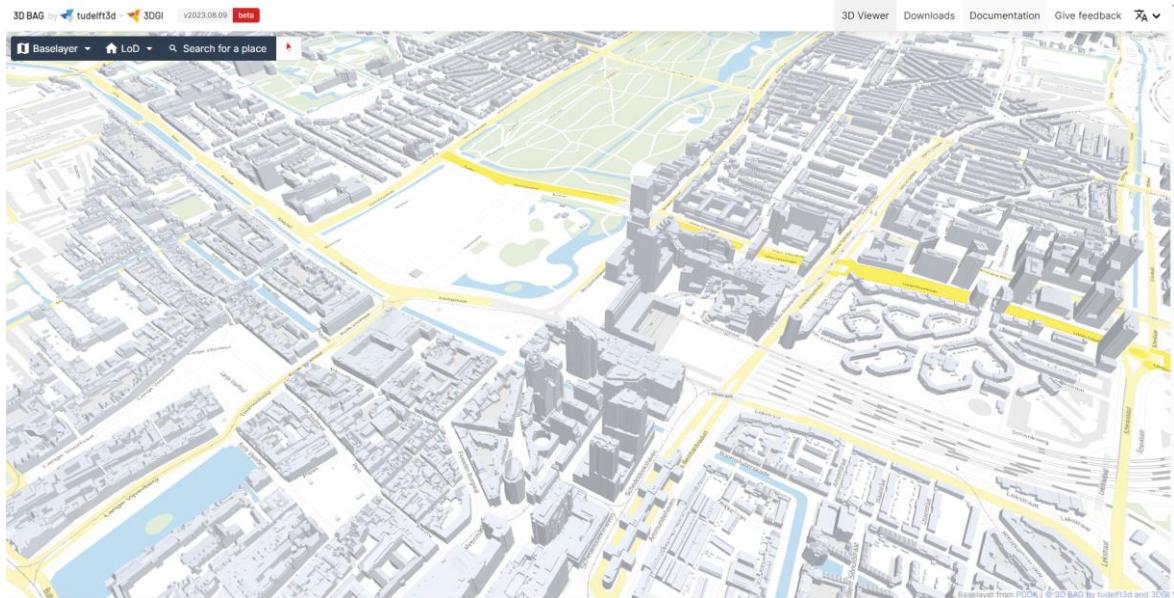
**CesiumJS** je knihovna umožňující tvorbu interaktivních 3D aplikací založených na geoprostorových datech. Zaměřená je převážně na zobrazení velkých datových sad jako jsou modely terénu, budovy a družicové snímky, a to i v globálním měřítku, zpravidla ve formátech 3D Tiles. Výhoda CesiumJS je její integrace s Cesium Ion platformou, což je cloudové řešení úložiště a sdílení dat. Momentálně CesiumJS nepodporuje WebXR API, tudíž imerzní VR v rámci prohlížeče není možné, ačkoliv vývoj v tomto směru probíhá není však zatím oficiálně součástí CesiumJS knihovny. (CesiumGS 2023b; 2023a; Espinosa 2023). CesiumJS využívá pro 3D vizualizace např. Švýcarská mapovací služba, která poskytuje 3D model budov, vegetace a terénu pro celé Švýcarsko jako otevřená data.

**VTS Geospatial** od *Melown technologies* je *tech stack*, který poskytuje způsob integrace různých zdrojů dat (3D modelů, výškových map, digitálních modelů povrchu, vektorových dat atd.) do jednotného výstupu, který je následně možné konzumovat skrze desktopové (vts-browser-cpp, Unity, ArcGIS) či webové klienty (vts-browser-js popř. obecné WebGL aplikace). Obdobně jako CesiumJS klient vts-browser-js nemá WebXR podporu, tudíž bez rozšíření podporuje pouze neimerzní VR vizualizaci.

Další porovnání za účelem vývoje nové aplikace zobrazení 3D modelů budov v Holandsku provedli (Peters et al. 2021). Porovnávali technologie s požadavky na síťový výkon a nízké výpočetní nároky na klienta. Jako nevhodné vyhodnotili CesiumJS (příliš výpočetně náročné) a MapboxGL (nízká specializace na 3D a nejasné OpenSource definice). V rámci výzkumu vznikla aplikace **3dbag-viewer** založená na three.js zobrazující data skrze formát 3D Tiles (budovy) a WMTS (podkladová mapa). Prohlížecí aplikace sama o sobě nepodporuje imerzní VR, ale jelikož je založena na knihovně three.js je zde potenciál pro rozšíření.

**iTowns** je framework založen na three.js, umožňuje vizualizaci geografických dat ve 3D prostředí podporuje geoprostorových služeb WMS, WMTS aj. a dat 3D Tiles, GeoJSON. iTowns nepodporuje integraci s WebXR. Na úrovni desktop VR umožňuje virtuální průchod obdobný, jaký poskytuje streetView na Google Maps. Sám o sobě framework nepodporuje rozšíření do imerzní VR úrovně, ale obdobně jako 3dbag-viewer je založen na three.js tudíž rozšíření o WebXR funkcionalitu je teoreticky možné.

**Deck.gl** je knihovna umožňující 3D vizualizaci geoprostorových dat. Jejím hlavním zámkem je vizualizace velkých datových sad v podobně vrstev, obdobně jako v tradičních GIS. Knihovna je zaměřena především na tematickou vizualizaci. Knihovna je kompatibilní s tradičními geoprostorovými formáty jak GeoJSON aj. Míra imerze je na úrovni Desktop VR, kdy knihovna umožňuje zobrazení na tradičních obrazovkách a interaktivitu v 3D prostředí pomocí myše a klávesnice tedy přiblžení a posun. Knihovna momentálně nepodporuje WebXR API tudíž rozšíření míry imerze je momentálně nemožné a knihovna není vhodným kandidátem. (deck.gl 2023)



Obr. 17 3dbag-viewer. zdroj: (Peters et al. 2021)

Tab. 5 WebGL knihovny se zaměřením na geoprostorová data.

Řešení	Prog. Jazyk	WebXR	Typ dat	Licence
ArcGIS + ArcGIS JS API	JS	Ne	Geografická data	Proprietární
City Engine + VR Export 360°	CGA, JS	Ano	Modelování měst	Proprietární
QGIS + qgis2threejs	Python, JS	Ne	Geografická data	GNU
Luciad Ria	JS	Ne	Geografická data	Proprietární
CesiumJS	JS	Ne	Instancovaná geografická data	Apache 2.0 (s výhradami)
vts-geospatial	JS, C	Ne	Geografická data	BSD
MapBoxGL	JS	Ne	Geografická data	Proprietární (Mapbox TOS)
deck.gl	JS	Ne	Tematická geografická data	MIT

Na základě výše uvedených řešení je možné tvrdit, že aplikace zobrazující geoprostorové informace dominuje především 3D Tiles formát a následně vykreslovací enginy cesium.js a three.js. V případě technologií zabývajících se zobrazením 3D scén na webu je možné tvrdit, že mají buďto minimální popř. žádnou podporu pro tvorbu virtuálních zážitků. Na základě této analýzy je tedy možné tvrdit, že tradiční geoprostorová řešení poskytují malou integraci s technologiemi umožňujícími virtuální realitu. Za účelem vývoje imenznho interaktivního VP, je nutné identifikovat vhodnější řešení.

### 4.3.2 Vykreslovací enginy

Jak již bylo zmíněno v kapitole 4.2 Webový vývoj drtivá většina 3D grafiky na webu je realizována skrze nízko úrovnovou knihovnu WebGL. Přímá práce s touto knihovnou je však není triviální, tudíž existují knihovny dedikované pro usnadnění vykreslování 3D grafiky na webu. Jedná se o knihovny, které zpravidla poskytují deklarativní objektovou strukturu nad WebGL koncepty. Poskytují tedy implementované abstrakce pro jednotlivé komponenty 3D vizualizace skrze zpravidla JavaScriptové API. V následující kapitole jsou vybrány hlavní zástupci této kategorie. Výběr byl vytvořen na základě popularity řešení a poslední aktualizace projektu. Projekty s poslední aktualizací starší než 3 měsíce nebyly brány v potaz.

**Tab. 6 Populární WebGL knihovny. Popularita – počet hodnocení na GitHub k datu (29.9.2023). zdroj: (Seguin 2023)**

WebGL knihovna	Popularita	Prog. jazyk	WebXR	Import	Funkcionalita	Licence
three.js	95 tis.	JS	ano	glTF, OBJ, FBX, COLLADA	Vykreslování, Interaktivita	MIT
Babylon.js	22 tis.	JS	ano	glTF, OBJ, STL, FBX, Babylon	Multifunkční, Interaktivita	Apache 2.0
PlayCanvas	8.7 tis	JS	ano	FBX, OBJ, glTF	Herní engine	MIT
filament	16 tis.	C++	ne	glTF, OBJ, STL	Vykreslování	Apache 2.0
Hilo3d	623	JS	ne	glTF, OBJ	Vykreslování	MIT
LayaAir	1.5 tis.	JS	ne	LayaAir, FBX, OBJ, glTF	Vykreslování	MIT
p5.js	20 tis.	JS	plugin	OBJ, STL (s doplňky)	Vykreslování	LGPL
oGL	3.3 tis.	JS	ne	glTF, OBJ (s doplňky)	Vykreslování	MIT

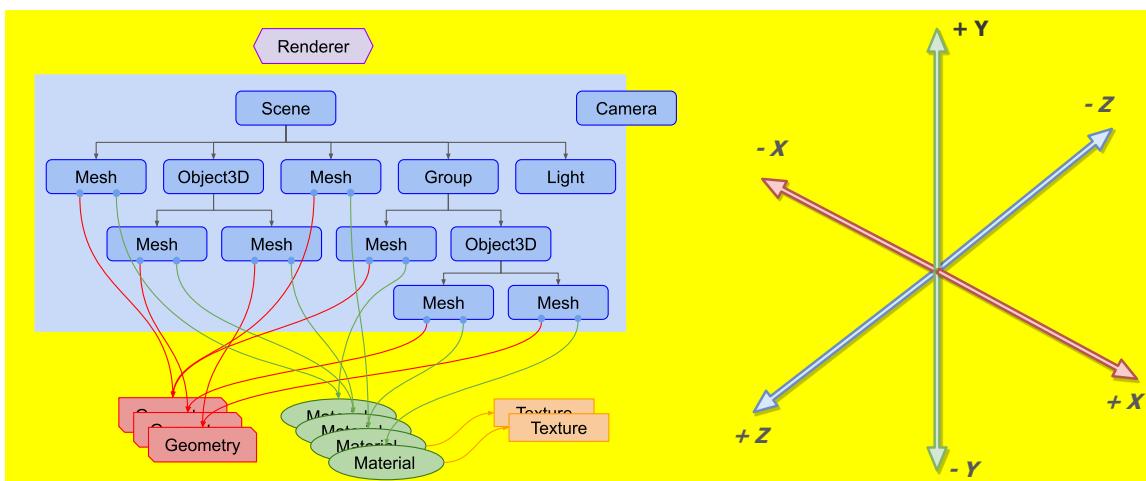
Pro další analýzu jako vhodné byly vybrány populární řešení **three.js**, **Babylon.js** a **PlayCanvas**, jelikož jako jediné poskytují přímou podporu pro WebXR API.

#### Three.js

Jedná se o Javascript knihovnu, tvořící abstrakci pro práci s WebGL. Knihovna byla vytvořena Ricardem Cabellem a je publikována pod MIT Licencí. Jedná se o velice populární knihovnu (více než 1.1 mil. instalací skrze npm), která je v aktivním vývoji. (npm 2023) Knihovna poskytuje velice detailní a udržovanou dokumentaci, širokou komunitu s řadou již existujících implementací. Z tohoto důvodu je three.js de-facto standard pro renderování 3D grafiky na webu (Meta Developers 2022). Three.js disponuje vlastním ekosystém frameworků a implementací, které rozšiřují její funkcionalitu. Patří mezi ně např.:

1. **r3f** (React Three Fiber) – Knihovna pro tvorbu interaktivních 3D uživatelských rozhraní pomocí frameworku React JS.
2. **A-frame** – Deklarativní systém pro tvorbu virtuálních prostředí a integraci s WebXR API.
3. **Ammo.js, Cannon.js** – Knihovny dodávající funkcionalitu fyzikálních simulací
4. **Needle engine** – integrace desktopových aplikací Unity, Blender aj.

Three.js poskytuje soubor tříd pro jednotlivé komponenty 3D vizualizace. Hlavním rozhraním je *Renderer*, který při poskytnutí Scény a Kamery umožní skrze WebGL vykreslit část 3D prostředí, které je v záběru kamery jakožto 2D obraz v rámci <canvas> HTML elementu. Hlavní strukturou v three.js je graf scény Obr. 18 Graf scény a kartézský souřadnicový systém three.js aplikace. (three.js Contributors 2023c; Discover three.js Contributors 2023).



Obr. 18 Graf scény a kartézský souřadnicový systém three.js aplikace. (three.js Contributors 2023c; Discover three.js Contributors 2023)

Geolokace v three.js není inherentně implementována. Využívá 3D kartézské soustavy kdy střed scény je ( $x = 0, y = 0, z = 0$ ) a jednotkou je metr. Jedná se o standard pro vykreslovací enginy a jiné 3D softwary. Orientace os je však častým problémem nekompatibility jak mezi softwarovými řešeními, tak i kartografickými souřadnicovými systémy. V three.js každý z objektů, který je v rámci scény má svůj lokální kartézský souřadnicový systém. TRS (*translace, rotace, scaling*) je definována v souřadnicovém systému rodičovského nádu v rámci grafu scény.

Three.js poskytuje abstrakci pro tvorbu VP, za účelem dosažení interaktivního VP je však nutné svět animovat, tedy vykreslit požadovaný počet snímků. Tato operace je v three.js nazývána *Animation loop*, pomocná, níž je vykreslování X snímků za Y časovou jednotku. Pro dosažení 60 FPS je potřeba vykreslit snímek každých 16 milisekund. Tato operace je velice podobná tzv. *Game Loop*. Termín pocházející z herního vývojářství popisující cyklus průběhu hry skrze operace: 1) Získání uživatelského vstupu, 2) Výpočtu simulací, 3) Aktualizace animací, 4) Vykreslení snímku. V případě vývoje aplikace pro virtuální realitu je nutné tuto animační smyčku vhodně synchronizovat s životním cyklem WebXR aplikace (viz. kap. **Error! Reference source not found. Error! Reference source not found.**). Propojení s WebXR životním cyklem je v three.js implementováno skrze třídu VRButton. Návratovou hodnotou třídy VRButton je html <button> element, který umožní spuštění VR relace. Třída dále zajišťuje korektní dotaz na typ virtuální relace. Při úspěšném dotazu na VR relaci, třída automaticky propojí vykreslovací smyčku three.js rendereru s výstupním zařízením. Three.js dále poskytuje předpřipravené 3D modely ovladačů populárních HMD zařízení. Za účelem tvorby imerzního prostředí je tedy zapotřebí vlastního vývoje kolizní logiky (kolizní geometrie, raycasting), pohybu ve scéně a celkově způsobu interakce s objekty. Z tohoto důvodu je vývoj VP jen za pomocí three.js netriviální záležitostí a je vhodné využít již implementovaná řešení.

## Babylon.js

Babylon.js je obdobně jako three.js javascriptová knihovna umožňující vykreslování 3D scén na základě WebGL aj. API na nižších úrovních. Babylon.js je komunitní open source projekt Babylon.js mimo vykreslovací funkcionalitu poskytuje i prvky herních enginů jakožto fyzikální simulace, animace, VR, raycasting aj. Babylon.js podporuje metody PBR (*physically based rendering*) vykreslování a výměnný formát glTF. Dále Babylon.js poskytuje nástavby nad jádrovou knihovnou ve formě online nástrojů jako je online editor kódu, inspektor 3D scén a vizuální editor pro vytváření materiálů. (WebXR 2020)

Babylon.js podporuje WebXR API v plné šíři specifikace, tedy VR i AR. Poskytuje abstrakce pro práci s WebXR ve formě ekvivalentních objektů v Babylon.js. Podpora pro dynamické prolínání mezi XR a desktop přístupem. Obdobně jako v three.js poskytuje VRButton třídu, skrze kterou je možné propojit vykreslovací cyklus s WebXR.

Interakce je nativně podporována a jsou poskytovány jednoduché abstrakce. Uživatelská selekce je implementována napříč zařízeními skrze *fallback* model. Kdy uživatelská selekce je mapována na aktuální dostupné zařízení, jimiž může být tlačítko na HMD ovladači, myš, popř. selekce pohledem. Pohyb je implementován primárně pomocí teleportace. Při potřebě pokročilé interakce je nutná vlastní implementace. Babylon.js je primárně knihovnou, tudíž tvorba scény stejně jako u three.js probíhá v rámci kódu.

### PlayCanvas

Jedná se o Javascriptové cloudové řešení s vlastní abstrakcí nad WebGL. PlayCanvas poskytuje grafický editor ve webovém prostředí a možnost tvorby vlastních interakcí a herní logiky skrze předpřipravené komponenty, popř. vlastní JS skripty. Z tohoto důvodu je možné jej zahrnout i do kategorie herních enginů. Velkým rozdílem od výše zmíněných knihoven je fakt, že se jedná o kompletní řešení, které poskytuje grafický editor pro tvorbu 3D scén, textový editor a vývojový server přímo v prohlížeči. Zároveň je možné výsledný projekt publikovat přímo z PlayCanvas editoru. Při tvorbě složitějších scén je omezený výkon webového editoru patrný.

### 4.3.3 VR frameworky

Technologie uvedené v předešlé kapitole jsou obecné přístupy k vykreslování 3D grafiky na webu. V případě tvorby VP je tedy nutné se zaměřit na technologie specializující se na tvorbu virtuálních prostředí. Jelikož virtuální realita vyžaduje vykreslování 3D prostoru, jsou technologie umožňující tvorbu VP často nadstavbou nad knihovnami zmíněnými v předešlé kapitole.

#### A - Frame

A-Frame je webový framework umožňující tvorbu virtuálních prostředí. A-Frame je založen na HTML. Jedná se o *entity-component framework*, který poskytuje deklarativní a rozšiřitelnou strukturu nad knihovnou three.js. A-Frame využívá three.js pro manipulaci WebGL primitiv. *Entity – Component* přístup umožňuje definování entit jakožto elementů přímo v HTML kódu a následně definování komponent v rámci JavaScriptu. Při renderování scény A-Frame knihovna vytváří hierarchii DOM prvků z HTML elementů, které představují různé objekty ve scéně. Tyto prvky mohou být vybírány a manipulovány pomocí JavaScriptu, stejně jako jakékoli jiné HTML prvky. Například lze pomocí JavaScriptu měnit pozici, rotaci nebo vzhled objektu ve scéně. A-Frame také poskytuje sadu vestavěných komponent, které lze přidávat k DOM prvkům, aby jim byly poskytnuty VR specifické vlastnosti, jako například schopnost reagovat na sledování hlavy, sledování ruky nebo dotykové události. Navíc mohou vývojáři vytvářet vlastní komponenty pro rozšíření funkčnosti. A-Frame využívá DOM jako základ pro vytváření a manipulaci s prvky VR na webové stránce. Jednoduše A-Frame vytváří framework, ve kterém

je možné o 3D prostředích přemýšlet jako HTML dokumentech. Z pohledu herních enginů aframe nahrazuje GUI editor strukturou HTML.

V případě kompatibility s WebXR specifikací A-Frame rozšiřuje abstrakce (VR Button) poskytované v rámci three.js o další funkcionality. Pouhým přidáním elementu <a-scene> do HTML dokumentu je možné dosáhnout VP s podporou pro 3 DoF pohyb.

V případě komplexnějších projektů nežli je jednoduchá scéna s malým počtem objektů se především umisťování jednotlivých objektů v prostoru VP stává velice nepřehledné. V porovnání s GUI editory, kde je možné návrh scény provádět graficky je přístup návrhu scény v rámci HTML dokumentu méně vhodný. Dále komplexní projekty vyžadují pokročilou optimalizaci (počet vykreslovacích dotazů, komprese textur, komprese geometrie aj.). A-Frame sám o sobě poskytuje pouze základní podporu optimalizačních procesů. Je tedy na vývojáři, aby tyto techniky implementoval.

### Needle engine

Jedná se o webový runtime pro 3D aplikace (Unity, Blender). Umožňuje propojení mezi Unity a webovým runtime rozhraním, tím že podporuje export scén, animací, lightmap aj. skrze glTF standard. Needle engine je možné nazvat *workflow managerem* umožňující propojení mezi interaktivními technologiemi jako je Unity popř. Blender a webovým prostředím. Needle primárně podporuje otevřený glTF standard a vyžívá postupu, kdy je možné v binární formě (.glb) obsáhnout celou scénu. Primární výhodou Needle enigu je možnost rychlých iterací při vývoji. Změny provedené v Unity editoru se okamžitě promítají ve vygenerované three.js aplikaci běžící na vývojovém serveru. Unity engine sám o sobě poskytuje export projektů do HTML5 a WebGL umožňuje export pomocí IL2CPP do C++ a následně do *WebAssembly*, což umožňuje spuštění výkonnostně náročných scén ve webovém prostředí. Toto je však umožněno na úkor rychlosti prototypování a iterací ve vývoji samotné scény. Výsledná webová aplikace je kompletně vykreslována pomocí three.js. Komponenty definované v rámci Unity Editoru jsou mapovány na vlastnosti a metody three.js třídy Object3D a grafu scény. Vlastní komponenty je možné psát pomocí javascriptu. Needle Enigne následně komponenty automaticky přeloží do C# ekvivalentu, tudíž je možné s nimi automaticky pracovat v Unity. Needle Engine poskytuje funkcionality ve 3 hlavních formách (needle-tools 2023): Souhrn předpřipravených komponentů a nástrojů, které umožňují tvorbu scény v rámci Unity Editoru, export vytvořené scény do glTF formátu a webové prostředí, které načítá glTF soubory a vykresluje je pomocí three.js.

Needle engine poskytuje předpřipravené komponenty pro tvorbu základní scény VP. Pokročilá interakce s objekty, popř. pohyb pomocí teleportace je nutné implementovat samostatně. Z hlediska optimalizace Needle engine umožňuje automatickou konverzi textur do formátu KTX2, čímž výrazně snižuje velikost scén. Každopádně Needle engine není optimalizován pro velké projekty v dokumentaci uvádí maximální velikost exportovaných projektu 50 MB a maximálně 500 000 vertexů, a to jen v případě že cílovou platformou je desktop.

#### 4.3.4 Herní enginy

Tradičně se jedná o desktopové aplikace specializované pro vývoj počítačových her, popř. interaktivních prostředí. Primárním zaměřením je vývoj her. Výstupem je tedy binární soubor vytvořený pro danou platformu / operační systém. V případě virtuální reality je možné vyvíjet prostředí přímo pro dané platformy jako např. Meta Quest aj. Ačkoliv se jedná o desktopové aplikace, jak bylo zmíněno výše existují způsoby, jakými je možné herní enginy zapojit od tvorby virtuální reality pro webové prostředí. Primárním způsobem, jakým je možné propojit

tvorbu v herních enginech s webem je export kompletních projektů do WebAssembly<sup>6</sup>, který pak interaguje s DOM a WebGL API, tedy umožnění spuštění scén ve webovém prostředí. Mezi populární řešení je možné řadit Unity, Unreal Engine a Godot. Další enginy jako CryEngine popř. Source engine nejsou kompatibilní s WebGL a HTML5.

**Tab. 7 Přehled populárních herních enginů. (UnrealEngine 2023; Unity 2023; Seguin 2023)**

Jmeno	Programovací jazyk	WebGL	WebXR	Import	Export	Licence
Unity	C#	.NET + WASM	Ano	FBX, OBJ, DAE, glTF, STL aj.	bin	Proprietary
Unreal Engine	C++	plugin	Ne	FBX, OBJ, MAX, BLEND, glTF aj.	bin	Proprietary
Godot	GDScript (podobný Pythonu), C#, C++	Ano	Ano	DAE, glTF, OBJ, JSON a další	bin, DAE, glTF a další	MIT Licence
Wonderland Engine	JS, TS	Ano	Ano	glTF, FBX, OBJ, PLY, DAE aj.	bin (WebAssembly)	Autorské poplatky

## Wonderland Engine

Wonderland engine je platforma pro tvorbu virtuální a rozšířené reality na webu v jejímž zakladatelem a hlavním vývojářem je Jonathan Hale (Wonderland engine 2023). Wonderland engine zahrnuje desktopový editor s obdobnou funkcionalitou a UI jako tradiční enginy jako Unity a Unreal engine, spolu s webovým runtimem založeným na WebAssembly exportu. V porovnání s WebAssembly exportem z Unity je však iterační doba vývoje ve Wonderland mnohem rychlejší, u jednodušších projektů takřka instantní. Wonderland engine je technologií, která je stále v úvodní vývojové fázi. Engine je vyvíjen od roku 2019 malým týmem 3 lidí. Jedná se tedy o místy nestabilní technologii, u které je možné očekávat nestandardní chování. Wonderland engine je proprietární *closed source* řešení, které umožňuje volné využití s podmínkou, že veškeré projekty vytvořené v rámci Wonderland enginu nepřináší roční zisk vyšší než 120 000 USD. V tom případě Wonderland požaduje licenční poplatek ve výši 10 % z ročního příjmu. Pro osobní, popř. akademické využití je Wonderland volně dostupným řešením. Wonderland obdobně jako Unity a A-Frame umožňuje tvorbu aplikací pomocí ECS architektury. Vlastní komponenty je možné psát v javascriptu. Skrze komponenty je následně možné dosáhnout složitějších forem interakce.

Na rozdíl od zmíněných obecných herních enginů je Wonderland specializuje na vývoj VP ve webovém prostředí. Wonderland engine poskytuje plnou integraci s WebXR API. Wonderland vhodně abstrahuje specifika životního cyklu WebXR aplikace, jako jsou dotaz na mód lokálního prostoru, požadavek na uživatelský vstup pro inicializaci VP a propojení s vykreslovacím cyklem, za jednoduchá nastavení v rámci editoru. Za účelem vývoje VP tedy v případě Wonderland enginu není nutné znát WebXR API. Tvorba jednoduchého VP spočívá pouze v návrhu scény (rozmístění objektů v 3D prostoru) a spuštění scény přímo z aplikace.

---

<sup>6</sup> WebAssembly (WASM) je binární instrukční formát poskytující virtuální stroj pro exekuci kódu v internetových prohlížečích. Tedy kód napsaný v jiném jazyce než JavaScriptu může být spuštěn v prohlížeči. Umožňuje webovým aplikacím dosáhnout výkonnosti téměř na úrovni nativního kódu.

Wonderland engine zkompiluje celý projekt do binárního WebAssembly souboru, který pak spustí na lokálním vývojovém serveru. (Wonderland engine 2023)

Primárním zaměřením enginu je odstranění manuální optimalizace modelů a aplikační logiky za účelem dosažení vhodného výkonu. Wonderland tedy poskytuje možnosti optimalizace přímo v rámci editoru, primárně se jedná o automatické procesy zjednodušování geometrie a komprimování textur do formátu *.basis / ktx2* přímo v rámci editoru. Zároveň pro běh aplikace Wonderland implementuje techniku slučování (*batching*), kdy spojuje co možná nejvíce objektů a jejich materiálů, za cílem minimalizovat počet vykreslovacích dotazů (*draw calls*). Wonderland dále poskytuje detailní dokumentaci. Zároveň mimo dokumentaci má Wonderland engine velice silnou a ochotnou komunitu v prostředí veřejného Discord serveru. Spolupráce s touto komunitou je klíčová k řešení problémů při vývoji. Ačkoliv je Wonderland mladým projektem, existují již uživatelské knihovny, které rozšiřují vestavěné možnosti samotného enginu. (Hale 2023)

#### 4.3.5 Služby

Softwarové řešení tvorbu virtuálního prostředí skrze službu. Jedná se o řešení, kdy za účelem vytvoření VP není vyžadován vlastní vývoj. Uživatel pouze interaktivně konfiguruje, popř. pouze se účastní virtuálního prostředí. Populárním řešením v této kategorii je služba Mozilla Hubs.

#### Mozilla Hubs

Jedná se o softwarové řešení vytvořené společností Mozilla. Umožňuje tvorbu kolaborativních virtuálních prostředí v rámci webového prohlížeče. Mozilla Hubs je vystavěna na základech WebRTC pro komunikaci a A-Frame, Three.js a WebGL pro tvorbu, vykreslení a interakci 3D scén. Součástí Mozilla Hubs je i *Spoke Editor*, což je GUI webová aplikace, umožňující interaktivní tvorbu virtuálních prostředí přímo v prohlížeči. Mozilla Hubs je primárně navržena za cílem vytvořit virtuální kolaborativní prostředí, tedy vhodné pro interakci více lidí, tj. ekvivalent virtuálních hovorů skrze aplikace jako Zoom, MS Teams aj. Hubs umožňuje přístup a pohyb ve VP v různých mírách imerze, tedy skrze tradiční obrazovku, myš, klávesnici, ale i skrze řadu HMD. Hubs neumožňují fyzikální simulace (detekce kolizí mezi objekty), vývoj pokročilých interakcí s prostředím a objekty mimo základní pohyb a manipulaci a komplexní animace. Zároveň Hubs nepodporují komprimované formáty a neposkytuje způsoby optimalizace geometrie ani textur. V případě komplexních projektů se jedná o nevhodné řešení. Hubs jsou vhodné řešení primárně pro prostředí vyžadující přítomnost více účastníků, jimiž mohou být virtuální učebny, muzea, galerie (Paradowski Creative 2022a), které nevyžadují velká množství dat.

#### 4.3.6 Optimalizační nástroje

Primárním cílem optimalizace je dosažení efektivnějšího přenosu dat přes síť a rychlejší načítání stránek. Na proces optimalizace je možné nahlížet ze dvou pohledů: **Modifikace samotných 3D modelu** a **Komprese**.

Modifikaci 3D modelů tvoří 3 hlavní procesy: optimalizace stromové struktury formátu, optimalizace geometrie a optimalizace textur. První kategorie zahrnuje odstranění prázdných nódů. Druhá kategorie se zaměřuje na zjednodušení geometrie a eliminaci nepoužívaných UV map. Třetí kategorie se věnuje snižování velikosti textur. V případě komprese se pak jedná o uložení dat v takovém formátu, aby byla minimalizována jeho velikost. Příklady procesů zahrnují kompresi textur pomocí formátů jako WEBP, KTX2. Pro geometrii jsou zkoumány procesy využívající formáty jako glTF a .glb, s důrazem na *bundling* (slučování meshů za účelem

snížení počtu vykreslovacích příkazů), *pruning* (odstraňování nepotřebné geometrie) a *flattening* (simplifikace stromové hierarchie). Komprese geometrie je prováděna pomocí kompresních formátů jako *draco* a *meshopt*.

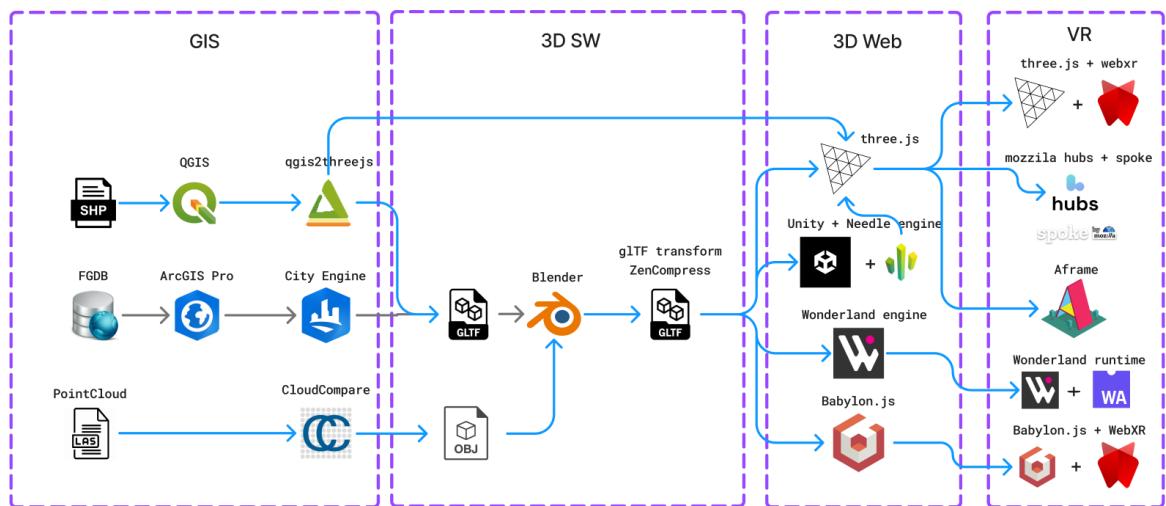
K provedení optimalizace je dostupná řada nástrojů.

1. **ZenCompress** nástroj určený pro kompresi 3D modelů, zaměřující se na efektivní snižování velikosti textur. Jedná se o desktopovou aplikaci s primárním zaměřením na kompresi textur do formátu .basis a ktx2. (Paradowski Creative 2022b)
2. **gltf-pack** a **gltf-transform**: Oba tyto nástroje jsou specificky navrženy pro manipulaci s formátem gltf. gltf-pack se zaměřuje na kompresi, zatímco gltf-transform poskytuje možnosti transformace i kompresi 3D modelů a textur. Jedná se o samostatné knihovny poskytující CLI a API pro javascript a python.
3. **Simplygon** a **Rapid Compact**: Jedná se o pokročilá komerční řešení problematiky optimalizace 3D modelů. Podporují řadu datových formátů nejen glTF a umožňují využití pokročilých algoritmů jak pro optimalizaci geometrie, tak textur.

Důležitým aspektem při výběru optimalizačních nástrojů je jejich podpora v různých prostředích a enginech. Například Mozilla Hubs nepodporuje formát KTX2 ve svém spoke editoru. Na druhou stranu řešení Wonderland Engine a Needle engine poskytují funkcionality zmíněných nástrojů v rámci svého procesu. Při použití komprimovaných modelů je pak nutné mít na paměti, že aplikace, která modely bude vykreslovat musí podporovat dekomprezii z daných formátů. V rámci této práce byla pro optimalizaci modelů využita kombinace manuální optimalizace v Blenderu, gltf-transform CLI a Wonderland enginu.

## 4.4 Výběr technologií

V následující kapitole je odůvodněn způsob výběru technologií pro implementaci pilotní aplikace.



Obr. 19 Soubor prakticky otestovaných technologií, modré – volně dostupný software, šedě – proprietární.

Výběr výsledné technologie proběhl na základě:

- Analýza existujících aplikací vyvinutých pomocí analyzovaných řešení.
- Tvorby testovacích aplikací ve většině výše zmíněných technologií. (viz. Tab. 8 Přehled testovacích aplikací.)

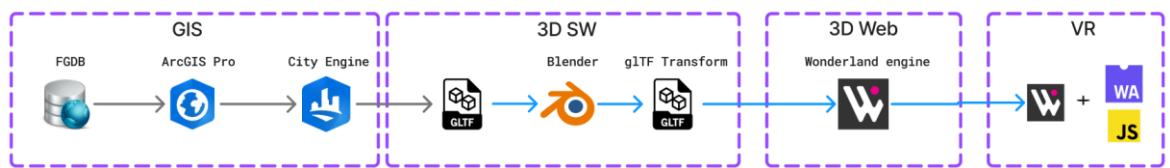
- Konzultace koncepce a zaměření technologie s členy komunity uživatelů / vývojářů.
- Zhodnocení vůči požadavkům plynoucím z charakteru geografických dat a jejich vizualizace.

**Tab. 8 Přehled testovacích aplikací.**

Technologie	Míra imerze	Pohyb	Interakce	URL
three.js	Imerzní VR	3 DoF	ne	<a href="https://jendahorak.github.io/disc3vr/">https://jendahorak.github.io/disc3vr/</a>
babylon.js	Imerzní VR	3 DoF	ne	<a href="https://jendahorak.github.io/babjs/">https://jendahorak.github.io/babjs/</a>
aframe	Imerzní VR	3 DoF	ne	<a href="https://jendahorak.github.io/a3sixty/">https://jendahorak.github.io/a3sixty/</a>
	Desktop VR	6 DoF	ne	<a href="https://foam-jumpy-dianella.glitch.me">https://foam-jumpy-dianella.glitch.me</a>
needle engine	Imerzní VR	3 DoF	ne	<a href="https://interesting-parallel-bit.glitch.me">https://interesting-parallel-bit.glitch.me</a>
mozilla hubs	Imerzní VR	6 DoF	ne	<a href="https://hubs.mozilla.com/v3xSqDE/obedient-high-sphere">https://hubs.mozilla.com/v3xSqDE/obedient-high-sphere</a>
	Imerzní VR	6 DoF	ne	<a href="https://hubs.mozilla.com/PFhZqGd/primary-stylish-festivity/">https://hubs.mozilla.com/PFhZqGd/primary-stylish-festivity/</a>
	Imerzní VR	6 DoF	ano	<a href="https://hubs.mozilla.com/CMa8Xah/vibrant-fixed-plane">https://hubs.mozilla.com/CMa8Xah/vibrant-fixed-plane</a>
wonderland engine	Imerzní VR	6 DoF	ano	<a href="https://jendahorak.github.io/wle-throw/">https://jendahorak.github.io/wle-throw/</a>

Výsledný rozhodovací proces je popsán následovně. Kapitola taxonomie technologií představuje řadu možností, jak je možné je dělit. Instrumentálním rozhodovacím kritériem je rychlosť a pohodlí vývoje aplikace. Primárním rozhodovacím kritériem byl fakt, zdali jsou technologie poskytují GUI editor scény. Tvorba komplexnějších 3D prostředí bez vizuálního editoru, je návrhově i implementačně značně náročná. Z tohoto důvodu byly vyřazeny základní technologie jako je **three.js** a **babylon.js**. Jejich přímá nástavba **A-Frame** již poskytuje širší abstrakci nad VP komponenty, pozicování a celková tvorba VP však probíhá v rámci HTML dokumentu. A-Frame poskytuje inspektor ve formě GUI 3D editoru, ten je však značně omezený a neposkytuje stejnou míru pohodlí při vývoji jako jiná řešení. Zbývající možnosti tedy jsou enginy jako **Wonderland engine** a **PlayCanvas** popř. propojení Unity s three.js skrze **Needle tools** nebo služby jejíž nejlepším zástupcem je **Mozilla Hubs**. Všechna tato řešení poskytují 3D editor pro tvorbu scén a různé úrovně předpřipravených komponentů pro tvorbu VP. Dalším hlavním rozhodovacím kritériem je zaměření na výkon, a tedy do jaké míry poskytují dané řešení optimalizační funkce. Tento aspekt je významný, jelikož geografická data obsahují velké množství geometrie. Nejslabším řešením v tomto ohledu je **Mozilla Hubs**, které neposkytuje takřka žádnou funkcionality ve formě automatické optimalizace modelů aj. V případě enginů v tomto ohledu je nejslabším řešením **PlayCanvas**, který neposkytuje optimalizační nástroje, obdobně jako u **Mozilla Hubs** je nutné optimalizaci provést externě. Jelikož je **PlayCanvas** plně cloudové řešení s editorem implementovaným v prohlížeči i editace a vývoj scén je díky slabému výkonu nekomfortním procesem. Ve výběru tedy zbývají **Wonderland engine** a **Needle tools**. Zde obě řešení poskytují možnosti automatické optimalizace (kompresi textur, geometrie aj.). **Needle engine** tento proces řeší automaticky při generování výsledné webové aplikace. **Wonderland engine** tento proces implementuje přímo v desktopovém editoru s tím, že poskytuje možnost volby kompresních formátů, popř. úrovně zjednodušení geometrie. **Wonderland engine** dále implementuje pokročilé metody optimalizace jako je cache, slučování textur do atlasů. Dalším plus pro Wonderland engine je celková koncepce řešení, jelikož se jedná o engine vyvýjen primárně pro tvorbu virtuálních prostředí na webu na rozdíl od **Needle engine**, který je více konverzní technologií mezi těžkými desktopovými 3D enginy a webovým prostředím. Zároveň Needle engine je zaměřen na obecné 3D prostředí, kdežto Wonderland engine je exkluzivně zaměřen na rozšířenou

realitu. Na základě těchto důvodů byl vybrán **Wonderland engine** jakožto, nejvhodnější řešení pro vizualizaci dostupných dat.



Obr. 20 Vybraný soubor technologií (*tech stack*) pro tvorbu virtuálního VP. modře – volně dostupný software, šedě – proprietární.

## 5 NÁVRH A IMPLEMENTACE VLASTNÍ APLIKACE

### 5.1 Návrh

Systematický přístup k návrhu tvorbu VP poskytuje (Çöltekin et al. 2020b), který dělí návrh XR aplikace na návrh vizualizace (*visualisation design*) a návrh interakce (*interaction design*). Stejný přístup byl využit v rámci návrhu pilotního VP v případě této práce.

V případě návrhu vizualizace je nutné brát v potaz specifika 3D vizualizace, které vedou ke specifickým návrhovým rozhodnutím. Použití 3D vizualizace je často považováno za špatné rozhodnutí v případě grafů, schémat a map, jelikož vizualizace může vést k informačnímu zahlcení a přebytečnou kognitivní náročnosti pro uživatele. Navíc v případě, kdy je vizualizace interaktivní a je po uživateli požadováno zapamatování a porovnání informací z paměti, není 3D vizualizace pozitivně korelována s lepšími výsledky. V případě kdy, jsou vyžadovány úkony jako je identifikace jevů, pojmenování objektů, rozpoznání podstaty nebo interpretace scény je 3D považováno za vhodné, jelikož poskytuje lidsky rozpoznatelnou kvalitu. (Çöltekin et al. 2020b). Dále autoři stejné publikace zmiňují, že významný vliv na návrh virtuálních vizualizací mají vliv tradiční vizualizační teorie (např. Bertinovy grafické proměnné) tak principy generalizace a LOD. Speciálně princip generalizace a LOD jsou velice významné v případě webového virtuálního prostředí, kdy značnou část návrhových rozhodnutí diktuje výkonnostní omezení a schopnost zahrnout velký rozsah zařízení. Návrhová rozhodnutí tedy v mnoha případech spadají pod rozhodnutí mezi vizuální kvalitou a dostatečným výkonem aplikace. Tuto myšlenku publikace rozvádí tím, že není vždy žádoucí maximalizace vizuální podrobnosti a realismu, ale existují situace, kdy je vyšší míra generalizace i přes dostupnost výkonu vhodným rozhodnutím.

Návrh interakce je následně soubor návrhových rozhodnutí, které obsahují způsoby, jakými uživatel interaguje s VP. Autoři (Çöltekin et al. 2020b) zmiňují, že klasické UI uzpůsobené možnostem tradičních vstupních zařízení není zcela aplikovatelné v XR prostředí. Virtuální prostředí jednak poskytuje pohyb v 3 dimenzích tak umožňuje nové způsoby interakce jako je chůze, uchopení a pohyb pomocí rukou aj. Tyto možnosti je tedy nutné zohlednit v návrhu aplikace. Způsoby interakce by měli být v souladu s účelem výsledného VP, zároveň nejběžnějším vstupním zařízením imerzní VR jsou ručně držené ovladače (Çöltekin et al. 2020b). Na základě tohoto tvrzení je tedy možné rozhodnout, že výsledná aplikace by měla primárně podporovat standardní vstup z nejvíce využívaných zařízení, jimiž jsou středně nákladné HMD s ručními ovladači. Zároveň se v tomto případě jedná o aplikaci webovou měla by tedy aplikace podporovat i přístup ze zařízení s nižší mírou imerze tedy tradičních desktopových displejů a mobilních telefonů. Zároveň je nutné následně interakci navrhovat tak, aby uživatelské úkony byly vykonatelné z různých zařízení, což např. při rozhodnutí podporovat mobilní telefony a desktop omezuje využití uchopení a přenosu jakožto způsobu interakce.

Dle výše zmíněných doporučení byl vytvořen postup pro vizualizaci prostorových dat v rámci webového VP. Následující kapitoly podrobně popisují návrhová a implementační rozhodnutí, a to nejdříve z hlediska vizualizace a následně z hlediska interakce. Vývoj aplikace je možné rozdělit na dvě hlavní etapy, a to předpřípravu dat (viz. kap. 5.2 Data a 5.3.1 Transformace dat) a vývoj samotné virtuální aplikace (5.3.2 Tvorba VP).

Obecnou metodikou využívanou v softwarovém inženýrství je specifikace uživatelských požadavků na výslednou aplikaci. Uživatelské požadavky jsou zpravidla děleny na funkční a mimo-funkční. V rámci této práce byla metodika specifikace požadavků byla inspirována pracemi (Herman 2014; Leitner 2020; Sháněl 2019) a modifikována na základě specifik

vyvíjené VR aplikace. Prioritizace požadavků proběhla dle MoSCoW metodiky (Miranda 2011), která dělí požadavky do následujících kategorií:

1. *Must have (M)* – požadavky bez kterých se aplikace neobejde, minimální možný set požadavků
2. *Should have (S)* – důležité požadavky, aplikace je funkční bez nich
3. *Could have (C)* – požadavky při jejichž vypuštění nedojde k žádné změně aplikace
4. *Won't have (W)* – požadavky které nebudou součástí implementace

Na základě této metodiky byla vytvořena matici požadavků, která byla rozdělena dle Tab. 9. Podrobný popis návrhových a implementačních rozhodnutí je popsán v kapitole 5.3.2 Tvorba VP. Zdali byl daný požadavek implementován v rámci výsledné aplikace je znázorněno pomocí sloupce Status a to na stupnici:

1. *Implemented (I)* – Implementováno v aplikaci
2. *Partially implemented (PI)* – Částečně implementováno
3. *Not implemented (NI)* - Neimplementováno

## 5.2 Data

Jak již bylo nastíněno v kapitole 3.5.2 Problém měřítka. Hlavním rozdělením, které je možné použít je dělení na statická a dynamická data (instancovaná) z pohledu načítání do scény na základě toho co uživatel vidí a v jaké je blízkosti k datům ve virtuálním prostoru. Z důvodu komplexity vývoje aplikace a malé dostupnosti instacovaných dat se tato práce zaměřuje primárně na způsoby vizualizace statických dat. Je tedy nutné brát v potaz limitace na velikost dat. Způsob konvertování statických dat do příslušného formátu ve vhodné velikosti vyžaduje použití řady softwarových řešení, což samo o sobě přináší míru komplexity v podobě kompatibility mezi danými řešeními. Následující kapitola se snaží najít vhodný postup a identifikovat a vyřešit případy snížené kompatibility.

Na základě dostupných dat města Brna Tab. 9 byly vytvořeny 3 statické 3D mapy. Proces jejich tvorby a specifika průběhu zpracování jsou popsána v následujících kapitolách.

**Tab. 9 Přehled datových vrstev pro vybrané vizualizace z hlediska formátů a datových modelů.**

Mapy	Datová vrstva	Výchozí reprezentace	Výchozí formát	Cílová reprezentace	Cílový formát
Tematické	Budovy 3D	PolygonZ	FGDB	Mesh	glTF
	Terén	Bitmapa	TIFF	Mesh	glTF
	Textura terénu	Bitmapa	PNG	Bitmapa	PNG / KTX2
Topografická	Budovy 3D	PolygonZ	FGDB	Mesh	Mesh
	Stromy	Vektorová data – Bod	SHP	Mesh	glTF
	Terén	Bitmapa	TIFF	Mesh	Mesh
	Textura terénu	Bitmapa	Služba	Bitmapa	PNG / KTX2

### 5.2.1 Specifika vstupních dat

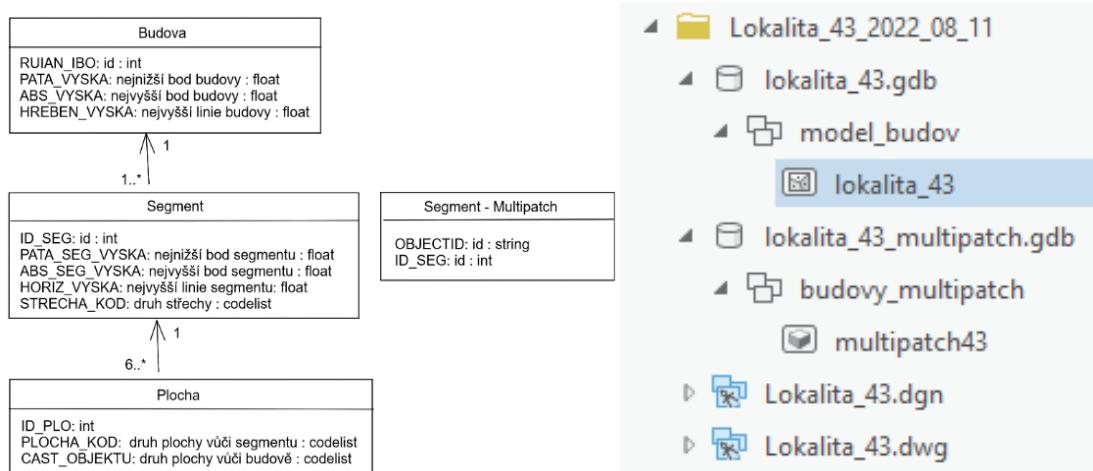
#### Budovy

Z důvodu důvěrné znalosti datové sady 3D modelu města Brna byla vizualizace v rámci práce založena na těchto datech. Tato kapitola se snaží na příkladu dat 3D modelu města Brna vysvětlit specifické problémy při užití primárně geografických dat určených pro práci v GIS do světa 3D vizualizace.

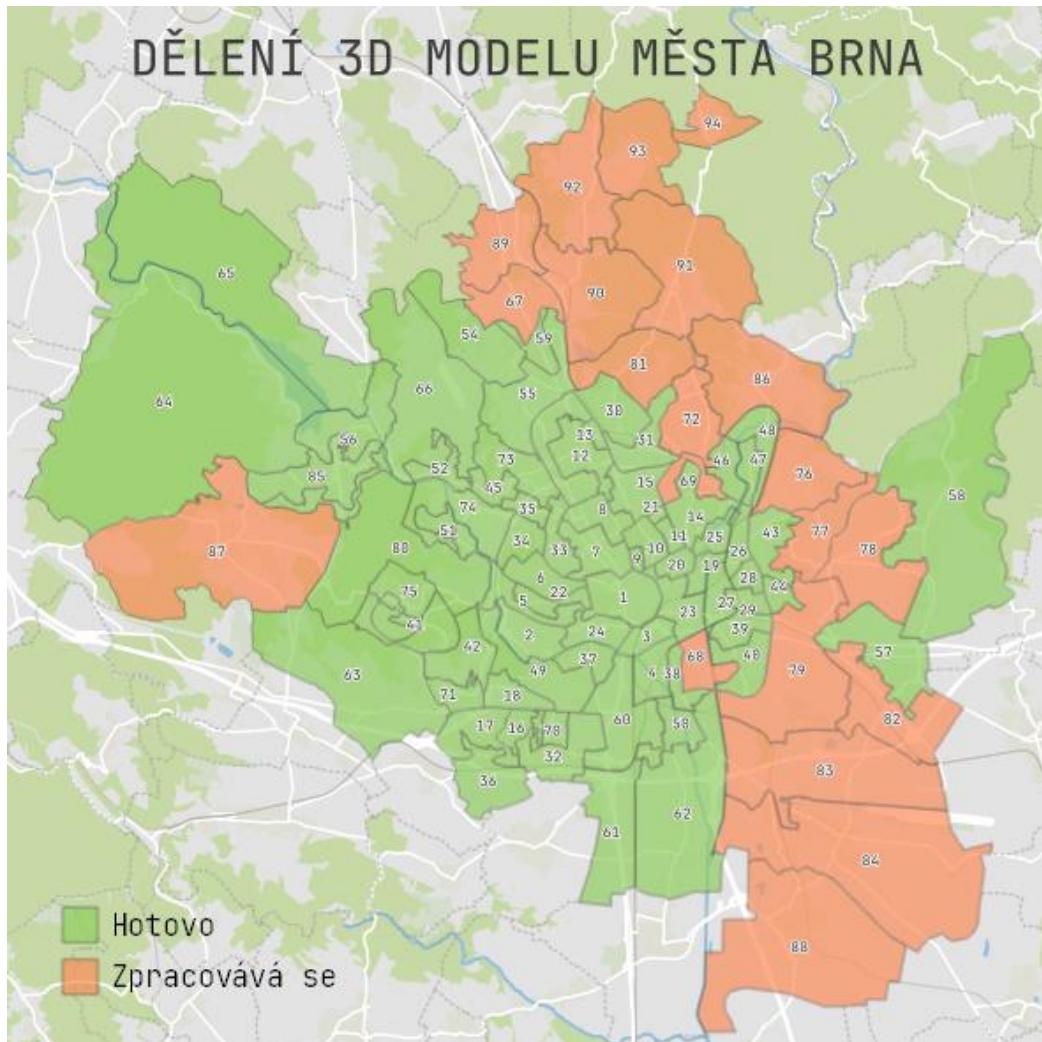
3D model budov města Brna je na zakázku magistrátu města Brna vytvářen firmou TopGIS. Jelikož se jedná o velké množství dat je model publikován ve specifické struktuře. Primárně je model dělen na lokality, po kterých je postupně vytvářen. Lokalita je také hlavní výdejní jednotkou. Pro každou lokalitu jsou poskytována data ve 3 formátech a různých mísách podrobnosti (GIS – FGDB – PolygonZ, Multipatch, CAD – DWG, DGN). V případě GIS formátů, je jedna lokalita jedna vrstva. V případě PolygonZ geometrie zahrnuje vrstva veškeré atributy zmíněné viz Obr. 21. V případě multipatch geometrie jsou prvky agregovány dle ID segmentu. Segmenty jsou definovány jakožto části budov se stejnou výškou římsy (hrana kde končí svislá stěna a začíná střecha). Model obsahuje 3 typy atributů a to:

- **Identifikační** (RUIAN\_IBO, ID\_SEG, ID\_PLO, OBJECTID)
- **Kvantifikační** (PATA\_VYSKA, ABS\_VYSKA, HREBEN\_VYSKA, PATA\_SEG\_VYSKA, HORIZ\_VYSKA)
- **Klasifikační** (STRECHA\_KOD, PLOCHA\_KOD, CAST\_OBJEKTU)

Klasifikační atributy byly vybrány pro tvorbu vizualizací. Pro které byla následně vytvořena metodika konverze z GIS do webového VP (viz. kap. 5.3.1 Transformace dat).



Obr. 21 Datový model 3D modelu města Brna (vlevo), složková struktura poskytovaných dat (vpravo).



Obr. 22 Dělení 3D modelu města Brna. zdroj dat: (MapTiler 2023)

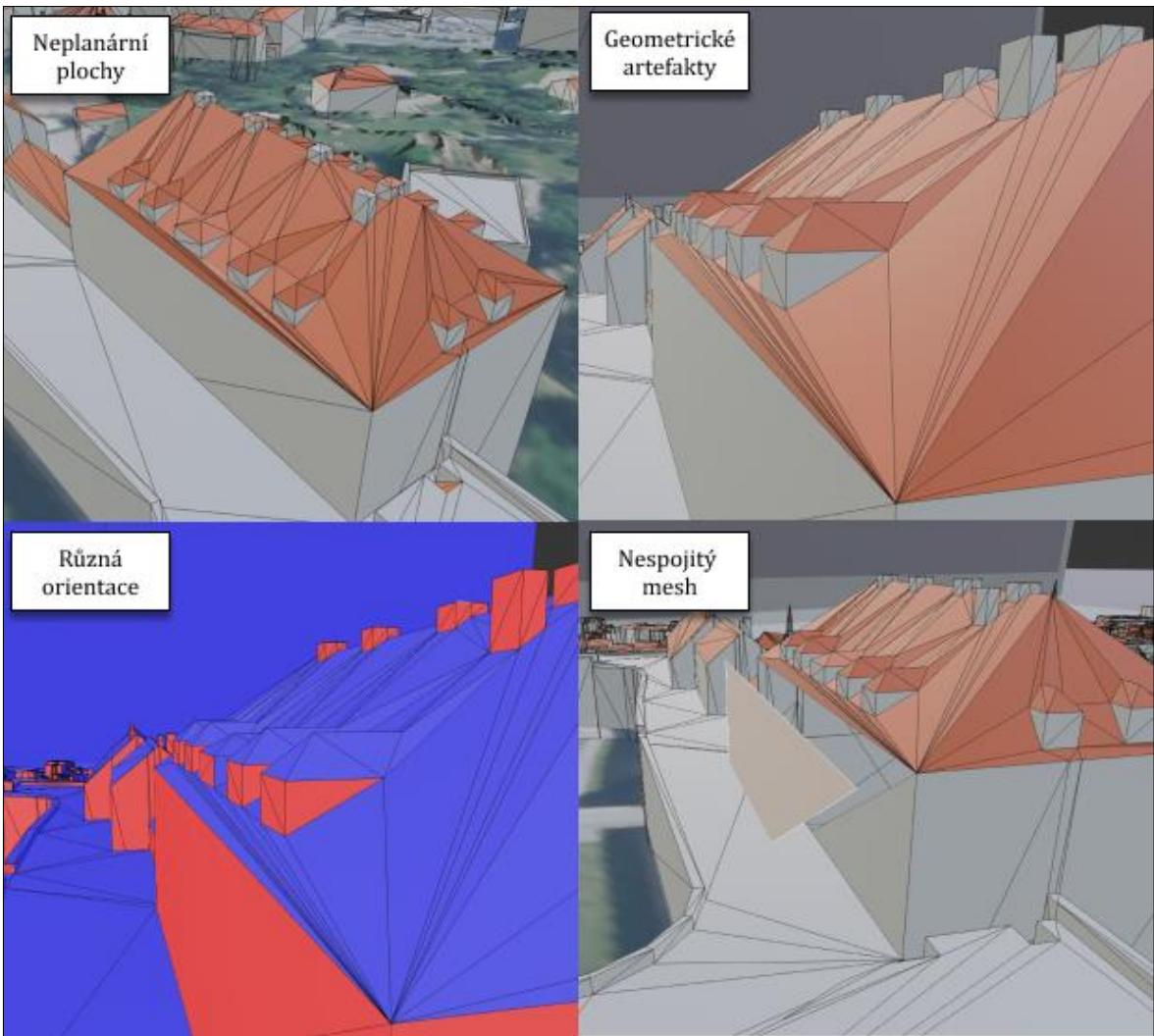
Hlavní problematikou vizualizace geografických dat poskytovaných v GIS formátech je způsob reprezentace dat. V GIS je symbologie definována pomocí vztahu Prvek – Atribut, kdežto v případě 3D vizualizace se jedná o vztah Mesh – Materiál. Zde je první problém, a to rozdíl v granularitě. V GIS je běžné zobrazit velké množství jednotlivých prvků se specifickými atributy, kdežto v případě vykreslování 3D grafiky je obecným pravidlem minimalizovat počet jednotlivých meshů a materiálů, které používají. Z výkonnostního hlediska je tedy nemožné pro každý GIS prvek vytvořit ekvivalentní mesh s přiřazeným materiálem. V případě zachování možnosti vizualizovat data na úrovni prvku dle atributů je nutné objevit způsob jakým je možné informaci o tomto vztahu přenést do 3D vizualizace. Standard glTF umožňuje definovat vlastní data až na úroveň vertexu, což by v případě převodu atributových se jeví jako vhodné řešení<sup>7</sup>. Vhodná cílová granularita je tedy definována počtem materiálů. Zde se tedy problém zjednoduší na otázku, jakým způsobem převézt vizualizaci z GIS softwaru do grafického softwaru jako např. Blender aj. ve formě meshů s přiřazenými materiály. Za účelem zachování sémantiky v podobě symbologie, tedy např. zobrazení geografických prvků klasifikovaných na základě vybraného atributu, je nutné uchovat informaci o tom do jaké kategorie patří daný prvek a jakou tato kategorie nese symbologii aby bylo následně možné dle kategorie definovat materiál a dle symbologie parametry materiálu.

<sup>7</sup> Práce na rozšíření glTF standardu o možnost zapisovat geoprostorová metadata je stále v průběhu. (Khronos Group 2022)

Dalším problémem je obecná kompatibilita mezi softwary. Tento problém se váže speciálně na data 3D modelu Brna, jelikož jsou poskytována v nativních Esri formátech. Za účelem práce s daty je tedy nutné mít přístup k ArcGIS Pro popř. ArcMap softwaru. ArcGIS Pro nativně poskytuje pouze modul *Export to Collada*, který byl na základě opakovaného testování vyhodnocen jako vysoce nevhodný, jelikož exportuje každý prvek jako samostatný .dae soubor a není možné zachovat definovanou symbologii. Vhodným nalezeným řešením je program Esri CityEngine, který je primárně koncipovaný jako software pro procedurální 3D modelování, každopádně umožňuje značnou podporu kompatibility mezi GIS a 3D vizualizací (Blender, Unreal, Hudiny atd.). City Engine umožňuje import nativních Esri formátů a export do glTF. Zároveň umožňuje tvorbu meshe pro DMR z bitmapy. Klíčovou funkcionalitou je pak možnost definování materiálů na základě atributů na úrovni prvků pomocí skriptovacího jazyka CGA. V případě otevřených formátů a softwarů obdobnou roli jako CE zastupuje QGIS plugin qgis2three.js, který umožňuje převádět definovanou symbologii do materiálu, triangulaci povrchů z bitmap a export do glTF formátu.

Třetí problém, který vyvstává z rozdílného pohledu na reprezentaci informace mezi GIS a 3D vizualizací je v tomto případě specifický pro 3D model města Brna. Jedná se o nežádoucí komplexitu dat, způsobenou metodikou tvorby dat. 3D model je vytvářen poloautomaticky na základě fotogrammetrických snímků. Hrany střech jsou anotovány lidskými operátory. Plochy střech, stěn a podstav jsou následně automaticky generovány. Tento přístup vede k 3 hlavním problémům.

1. **Neplanární plochy** – Plochy jsou generovány jako neplanární n-úhelníky tudíž při zobrazení v opravdovém 3D enginu dochází k automatické triangulaci a tvorbě přebytečných ploch. V některých případech se jedná o velká množství úzkých geometrií (geometrických artefaktů), což vede k výpočetní náročnosti při vykreslování.
2. **Duplikátní geometrie** – Budovy jsou poskytovány jako soubor jednotlivých ploch. Pro rohový bod budovy ve tvaru krychle data obsahují 3 separátní vertexy, nejedná se tedy o spojitý mesh. Což je výsledek toho, že každá plocha je samostatný prvek s vlastními atributy.
3. **Nekonzistentní orientace ploch** – Normálové vektory ploch nejsou orientovány jednotně, jelikož se nejedná o uzavřené objekty (problém č. 2). Tento problém zabraňuje jednoduché aplikaci materiálu na povrchy.



Obr. 23 Problematické aspekty 3D modelu Brna ve vztahu k 3D vizualizaci.  
zdroj: vlastní tvorba

Možná řešení na výše odvedené problémy jsou prozkoumány v rámci následujících kapitol, popisujících exemplární řešení pro 3 mapy 3D mapy vytvořené na základě vizualizace atributů (PLOCHA\_KOD, STRECHA\_KOD, CAST\_OBJEKTU).

### Terén

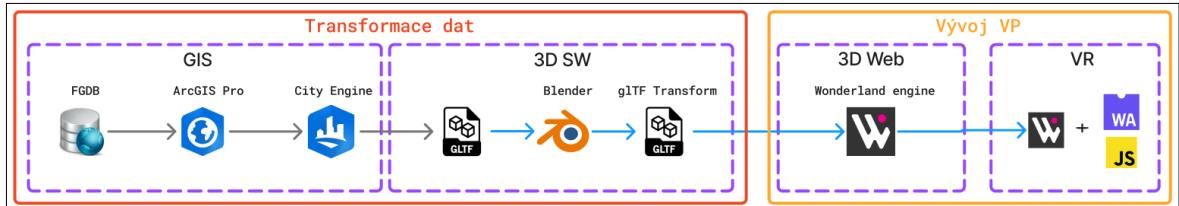
Hlavním procesem při převodu DMR popř. jakékoliv jiné povrchové reprezentace je triangulace na základě vstupní bitmapy. Tuto funkcionality poskytuje řada řešení. City Engine i qgis2threejs v rámci GUI rozhraní. Generovat terén je ale možné i v rámci webových vykreslovacích enginů Three.js a Babylon.js skrze *displacement mapping*. Klíčová charakteristika vygenerovaného terénu je míra podrobnosti geometrie, tedy jak moc podrobný bude výsledná síť reprezentující terén. Nástroje poskytují různou míru modifikace podrobnosti terénu. V rámci vypracování bylo využito programu CE, kde podrobnost terénu je definována podrobností (velikostí pixelu) vstupní výškové bitmapy.

### Podkladové mapy

Vizualizace terénu spočívala v potažení terénu texturou ve formě podkladové mapy. V případě topografické mapy se jednalo o ortofoto České republiky (ČÚZK 2023b). V případě tematických map pak byl podklad vytvořen v prostředí QGIS za pomocí podkladových map MapTiler (MapTiler 2023).

## 5.3 Vývoj aplikace

Následující kapitola je stěžejní kapitolou praktické části, jelikož popisuje proces transformace prostorových dat z GIS prostředí do prostředí virtuální reality na webu.

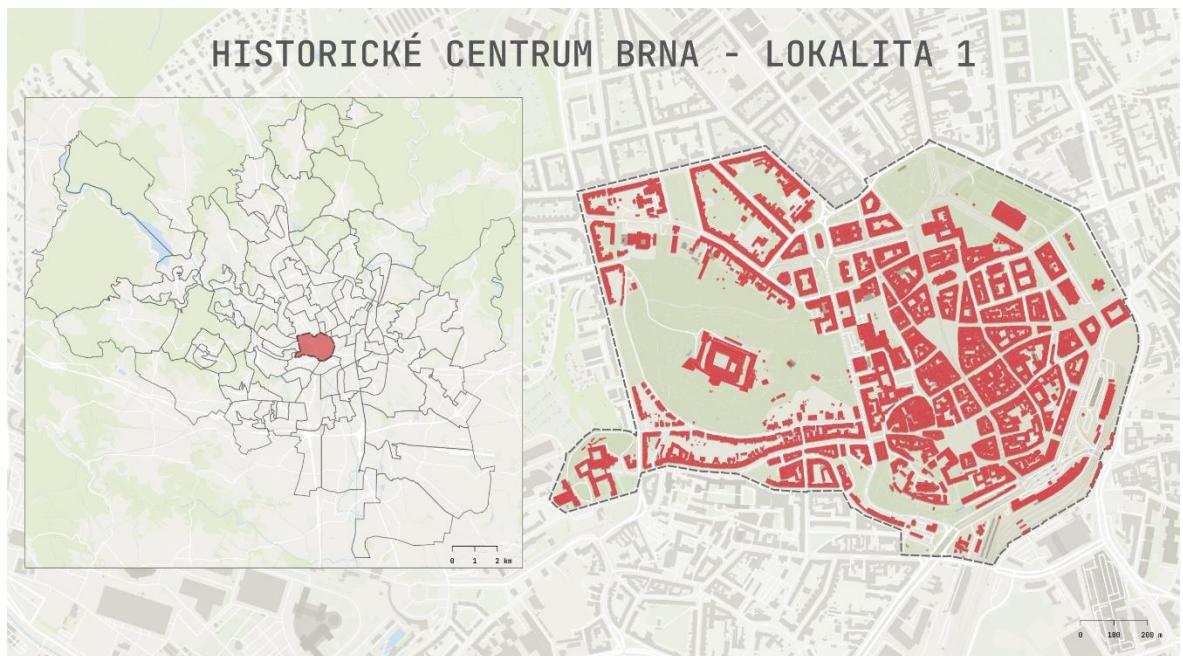


Obr. 24 Rozdělení procesu vývoje aplikace na přípravu dat a vývoj samotného virtuálního prostředí, dle použitých sw. řešení. zdroj: vlastní tvorba

### 5.3.1 Transformace dat

Klíčovou kapitolou v rámci práce popisující postup převodu vizualizací z GIS prostředí do webového VP. Zvolené klasifikační atributy byly vizualizovány v podobě 2 tematických map a jedné mapy topografické, popř. realisticky vypadajícího 3D modelu zvoleného území.

Pro testovací vizualizace byla vybrána oblast historického centra Brna. Přesné vymezení bylo zvoleno jakožto lokalita č. 1 dle řazení vyhotovení 3D modelu Brna (viz. předchozí kapitola). Tato oblast obsahuje výrazné vertikální dominanty (Petrov, Špilberk) a zároveň rozmanitou zástavbu.



Obr. 25 Zájmová lokalita pro tvorbu vizualizací. zdroj: (MapTiler 2023)

Kompletní postup převodu vybraných datových vrstev je uveden v diagramu viz. Příloha č. 2 Proces transformace dat. Klíčové aspekty postupu převodu vrstev jsou podrobně popsány níže. Diagram zobrazuje převod vrstev pro exemplární topografickou mapu. V případě tematických se jednalo o podmnožinu těchto vrstev. Jediným výrazným rozdílem, mezi mapami, který není v diagramu zohledněn je tvorba podkladové mapy pro tematické vizualizace, která proběhla v rámci prostředí QGIS a následně adekvátně namapována jakožto textura terénu v Blender prostředí. Relevantní části diagramu jsou prezentovány i v následujícím textu.

## ArcGIS

Jelikož formáty Multipatch ani PolygonZ nepodporují ukládání symbologie, je nutné parametry vzhledu uložit v rámci atributů jednotlivých prvků a v prostředí CE je namapovat na materiály přiřazené dané části meshe. Za dosažením tohoto cíle byl tedy rozšířen datový model 3D budov o atributy *colorCategory* (jméno materiálu / kategorie) a *colorValue* (hex kód vybrané barvy). Řešení bylo implementováno pouze pro jednoduché symboly, tedy pouze základní barvu (*Diffuse color* parametr materiálu).

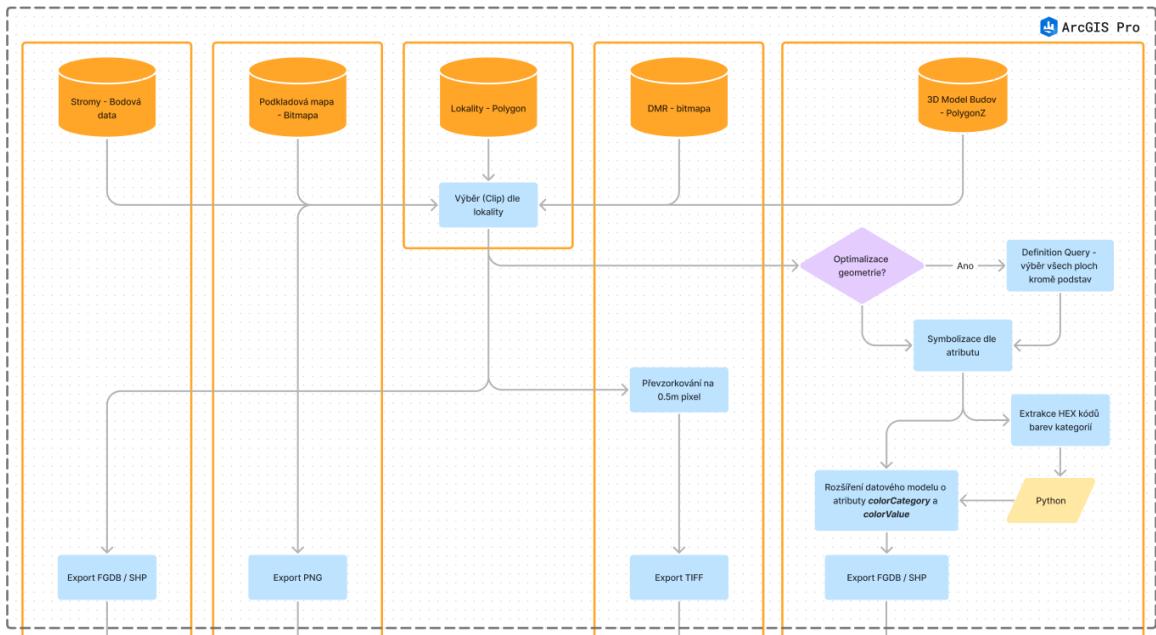
Při průzkumu a vývoji tohoto postupu byla objevena neshoda mezi datovými modely 3D modelu jednotlivých etap vyhotovení. Lokality vyhotovené v roce 2020 měli jiný datový model nežli etapy následující. Právě atributy umožňující sémantickou kategorizaci jako jsou STRECHA\_KOD, PLOCHA\_KOD a CAST\_OBJEKTU měly rozdílné domény a v daných lokalitách byly nevhodně přiřazeny. Jelikož je 3D model budov objemnou datovou sadou, která je rozdělená do specifické složkové struktury bylo za účelem sjednocení pod jedním datovým modelem, nutné úpravy provézt programicky, byl tedy napsán python script, který datové modely sjednotil a opravil přiřazení atributů na základě geometrie jednotlivých ploch<sup>8</sup>. Dále byl napsán skript za účelem mapování hodnot zmíněných kategorizačních atributů na jméno materiálu a HEX hodnotu zvolené barvy<sup>9</sup>. Nástroj byl napsán tak aby byl rozšiřitelný o libovolné mapování nových atributů, popř. změnu barevné stupnice zvolené pro vizualizaci. Skript je možné jednoduše použít i na specifickou složkovou strukturu 3D modelu, je tedy možné takto klasifikovat všechny lokality. Tato funkcionalita je žádoucí, jelikož zobrazení velkého množství dat v řešeních jako CE, Blender a hlavně webových vykreslovacích enginech je problematické, tudíž je nutné postupovat po menších částech.

Výsledným výstupem je tedy datová vrstva vybrané lokality rozšířená o atributy *colorCategory\_{zkratka řídícího atributu}*, *colorValue\_{zkratka řídícího atributu}* v případě PLOCHA\_KOD tedy *colorCategory\_pk*. Z ArcGIS byl proveden export dat v rámci lokality č.1 a to: vrstva budov rozšířená o vizualizační atributy, terén (bitmapa), podkladová mapa – ortofoto (bitmapa) a bodová vrstva stromů. Vše bylo exportováno jakožto FGDB a načteno do CE.

---

<sup>8</sup> Po dokončení všech lokalit 3D modelu bude skript využit na kompletní sjednocení 3D modelu a jeho následnou publikaci veřejnosti ve spolupráci s GIS oddělením Magistrátu města Brna a Kanceláří architekta města Brna.

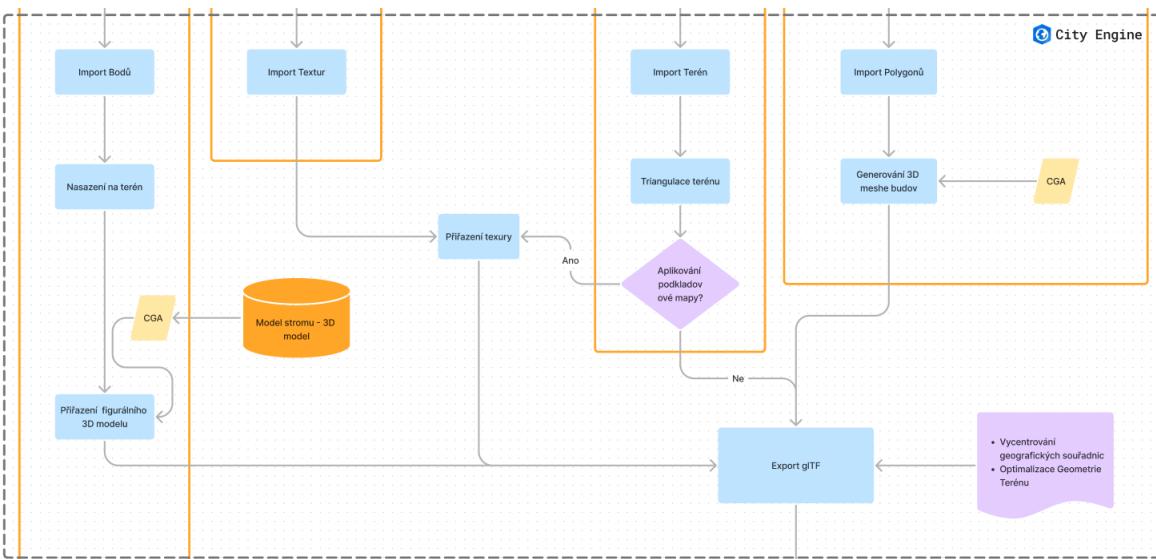
<sup>9</sup> [Zdrojový kód](#).



Obr. 26 Proces převodu dat. část 1. - ArcGIS Pro, zdroj: vlastní zpracovní

### City Engine

Software City Engine (CE) umožňuje převod geoprostorových dat (ESRI formátů) do formátů využívaných v 3D grafice. V rámci této práce je City Engine využit pro generování terénu z DMR rastrové bitmapy a k transformaci meshe ve formátu Multipatch na formát glTF, s tím že zůstane zachována symbologie na úrovni prvků. Generování DMR bylo provedeno pomocí vestavěných funkcí CE. Mapování symbologie na materiál a mesh pak skrze deklarativní jazyk CGA. Následně CE poskytuje exportér do gltf formátu, který umožňuje volbu obsahnut jak terén, tak generované modely.



Obr. 27 Proces převodu dat. část 2. - City Engine

### Blender

Zde vystává otázka, proč do procesu zahrnovat Blender, když CE umožňuje exportovat do glTF? Export z CE do glTF není dokonalý a CE neumožňuje podrobnou kontrolu optimalizace 3D dat, která je pro následné využití ve VP klíčová. Ačkoliv CE poskytuje možnosti optimalizace, bylo zjištěno, že tyto nástroje neřeší specifika vybraných dat. Blender je vhodným nástrojem

pro návrh výsledné scény při tvorbě VP, jelikož poskytuje pokročilé editační možnosti v rámci uživatelského rozhraní. Přidáním Blenderu do procesu zpracování dat, však přináší další problémy vzniklé na základě kompatibility softwarů.

V tomto případě se jednalo o rozdílnou implementaci BSDF shaderu a rozdílného barevného modelu, což mělo za důsledek, že importované modely měli výrazně jiný vzhled než v CE. Jakožto řešení bylo tedy nutné pro každý materiál vytvořit kopii s validním shaderem a korektně přiřazeným HEX kódem. Tento proces byl vyřešen kombinací python scriptů<sup>10</sup> s využitím Blender API a ruční editace. Klíčovým krokem pak bylo vyřešení různé orientace geometrie, tento proces byl vyřešen poloautomaticky. U variant vizualizace PLOCHA\_KOD a CAST\_OBJEKTU bylo nutné zapotřebí značné množství manuálních úprav. Mimo vyřešení problémů vzniklých ne-kompatibilou, byl Blender využit primárně pro optimalizaci modelu. Jednalo se o úkony:

1. **Zjednodušení grafu scény** – Tento krok spočíval v odstranění prázdných nódů z grafu scény, což zjednodušuje následnou práci ve webovém prostředí, jelikož není nutné traverzovat prázdné objekty.
2. **Vytvoření spojité meshe** – Jak pro terén, tak pro budovy bylo za využití nástroje *Merge by distance* vytvořen spojity mesh.
3. **Sjednocení objektů budov pod jeden Mesh objekt.** – Sjednocení objektů za účelem snížení počtu vykreslovacích příkazů.
4. **Zjednodušení geometrie** – Primárně pro terén byl použit *Decimate geometry* modifikátor, který sníží podrobnost dané sítě.
5. **Zmenšení textur** – Textury exportované z CE dosahovali rozlišení až 14000x14000, což je nepřípustná hodnota. Při existenci textury tedy byly tedy zmenšeny do rozlišení 4096x4096.
6. **Oprava orientace povrchů** – Bylo nutné předefinovat směr normálových vektorů jednotlivých ploch. Proces nebylo možné plně automatizovat, byla tedy nutná manuální úprava.

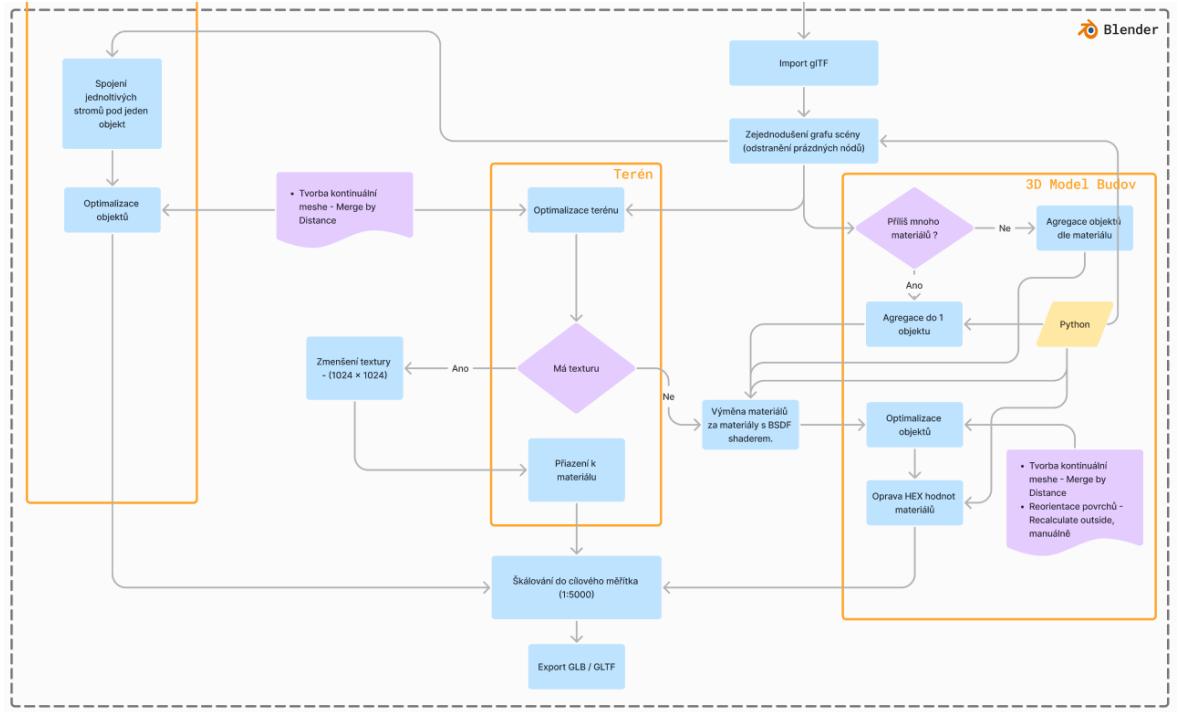
Nutno zmínit, že v případě optimalizace je vhodné zvolit optimalizační kroky v závislosti na výsledné implementaci. Při provedení následujících operací je možné model, popř. kompletní scénu exportovat a následně načíst do výsledné scény v rámci vybrané technologie. Za účelem dosažení vyšší míry optimalizace, a tedy možnosti zobrazení více dat při zachování dostatečného výkonu je vhodné použít dedikované nástroje pro optimalizaci jako zmíněný *gltf-transform*. Tyto procesy je však nutné koordinovat s možnostmi cílového vizualizační technologie. Především při použití komprese je nutné zajistit kompatibilitu v cílové technologii. V případě tematických map byla značná část optimalizace provedena až ve Wonderland enginu.

Dalším z klíčových kroků, bylo určení výsledného měřítka vizualizací. Na základě úvahy uvedené v kap. 3.5.2 Problém měřítka a na základě testování vizualizacích různých měřítek ve VP z hlediska příjemnosti práce s daty a výkonu aplikace, bylo zvoleno měřítko 1:5000. Do tohoto měřítka byly škálovány všechny 3 exemplární vizualizace. Výsledné 3D mapy<sup>11</sup> byly exportovány jakožto GLB a následně importovány do Wonderland enginu.

---

<sup>10</sup> [Zdrojový kód](#).

<sup>11</sup> Výsledný Blender projekt je možné získat skrze: [Projekt](#)



### 5.3.2 Tvorba VP

Tab. 10 Systém číslování uživatelských požadavků dle komponent aplikace. zdroj: vlastní tvorba

Typ	ID	Název
Funkční	A	Tematická mapa 1 – Typy střech
	B	Tematická mapa 2 – Části objektů
	C	Topografická mapa – Fiktivní Sochy
	D	Prostředí
Mimo-funkční	E	Mimo-funkční požadavky aplikace

#### Prostředí

Velká míra funkcionality definované v Tab. 11 byla z velké části umožněna díky vestavěné funkcionality Wonderland enginu (kolizní systém, kliknutí aj.) a knihovně wle-pp (Ducceschi 2023), což je knihovna komponentů rozšiřující základní funkcionality.

Tab. 11 Matice požadavků, D – funkční požadavky VP

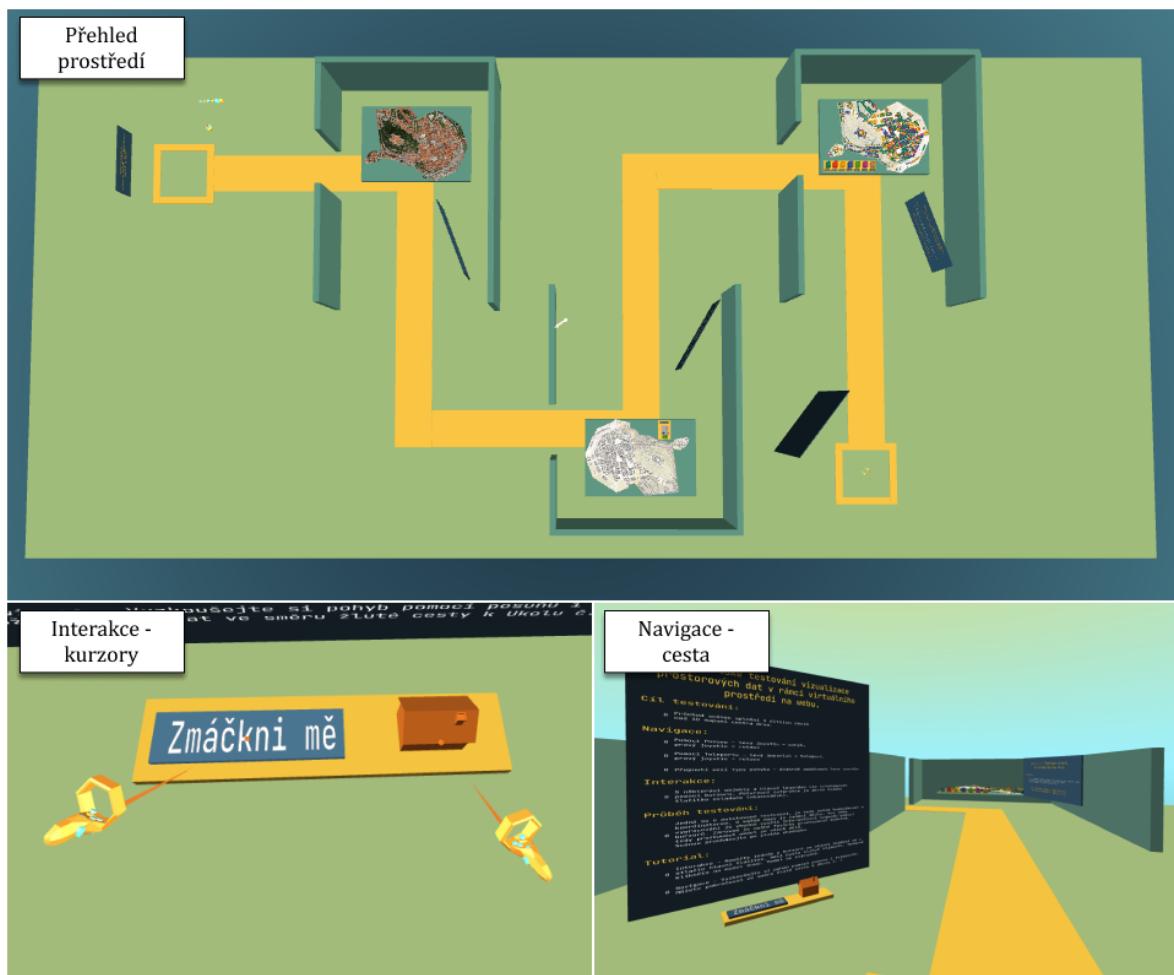
ID	Požadavek			Poznámka	Priorita	Status
D.D	Pohyb					
	D.D.1	Pohyb pomocí posunu		Aplikace umožňuje plynulý pohyb 3D prostorem.	M	I
	D.D.2	Pohyb pomocí teleportu		Aplikace umožnuje pohyb pomocí teleportace.	S	PI
D.E	Navigace					
	D.E.1	Intuitivní prostředí		Návrh prostředí jednoznačně vyjadřuje směr pohybu prostředím.	S	I
	D.E.2	Textová navigace		Aplikace obsahuje uživatelské instrukce v podobě textových boxů	M	I
D.F	Interakce					
	D.F.1	Binární vstup		Aplikace umožňuje funkcionalitu "kliknutí"	M	I
	D.F.2	Kurzor		Grafický kurzor zobrazující možnost interakce s prostředím.	M	I
	D.F.3	Kolizní systém		Uživatele je možné blokovat neprůchodnými objekty.	M	I
	D.F.4	Uchopení		Uživatel je schopen uchopit objekty a manipulovat s nimi.	W	PI
	D.F.5	Haptická odezva		Možnosti interakce jsou komunikovány skrze vibrace ovladačů.	S	I
	D.F.6	Zvuková odezva		Možnosti interakce jsou komunikovány skrze zvukové efekty.	S	I

D.D – Pohyb v rámci prostředí byl dvou modální. Primárním pohybem byl zvolen plynulý posun. Tento způsob je však znám tím, že může působit nevolnost při pohybu v rámci HMD

byla implementována i možnost teleportu. Možnost teleportu je dostupná pouze při přístupu skrze HMD s 6 DOF ovladači.

**D.E** – Navigace v rámci virtuálního prostředí byla vyřešena pomocí „chodníku“ a textové navigace.

**D.F** – Interakce byla vyvinuta na základě kolizního systému, který Wonderland engine poskytuje. Za účelem dosažení jednoduchého ovládání byl ovladačům přiřazen komponent *Kurzoru*. Tento komponent bylo nutné modifikovat tak aby dynamicky ukazoval kam uživatel míří pomocí geometrie vycházející přímo z ovladačů a geometrie kurzoru, který se objeví na objektu, s nímž je možné interagovat. Interakce s objekty dále vyvolá haptickou a zvukovou odezvu. Vyvinuté prostředí obsahuje i možnost interakce s objekty pomocí uchopení. Tato funkcionality však nebyla využita do plné míry.



Obr. 29 Prostředí, interakce a navigace.

## Tematická mapa 1 – Typologie střech

Tab. 12 Matice požadavků – A – Tematická mapa 1 - Typologie střech

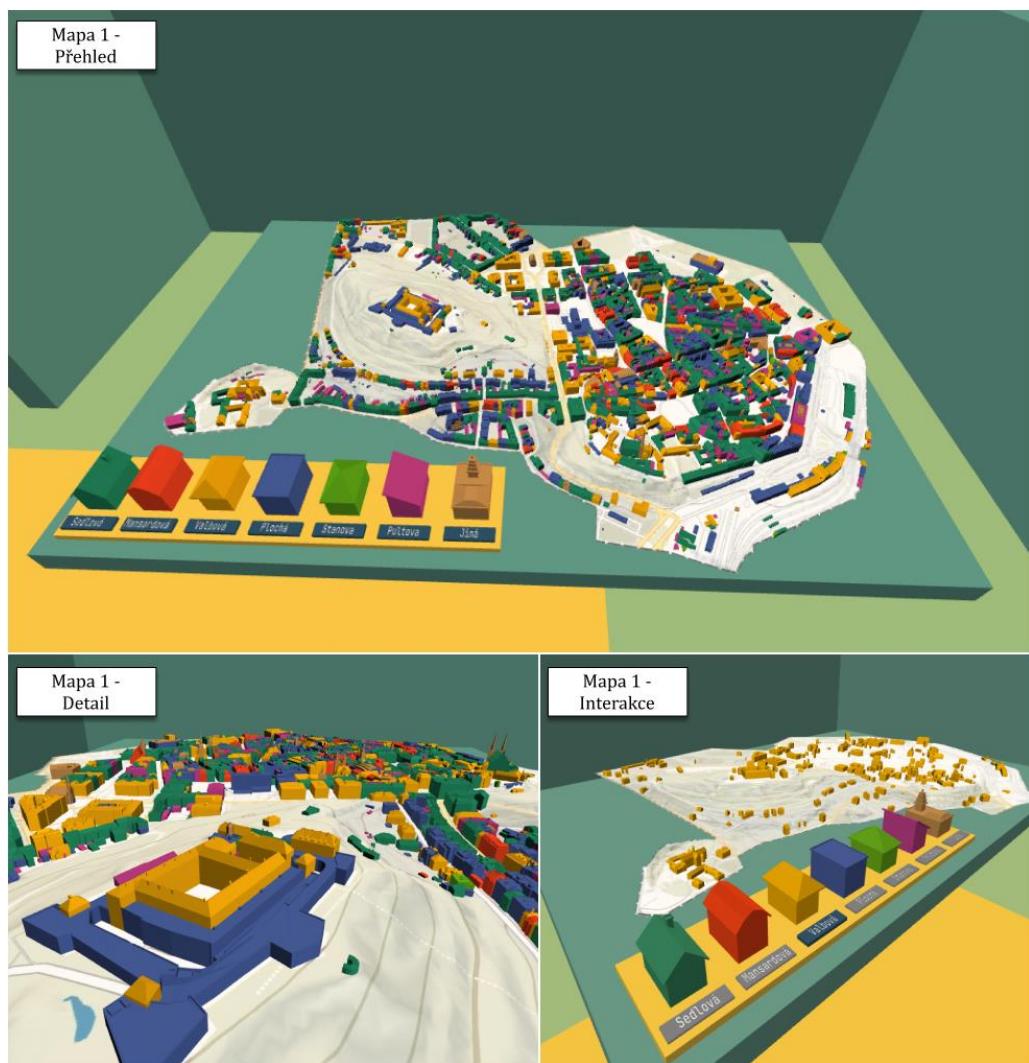
ID		Požadavek	Poznámka	Priorita	Status
<b>A.A</b>		<b>Vizualizace mapového pole</b>			
A.A	A.A.1	Zobrazení budov	Zobrazení vrstvy 3D budov.	M	I
	A.A.2	Zobrazení terénu	Zobrazení vrstvy 3D terénu.	M	I
	A.A.3	Texturování terénu	Zobrazení 3D terénu s texturou.	M	I
<b>A.B</b>		<b>Vizualizace legendy</b>			
A.B	A.B.1	Legenda ve 3D	Legenda k vizualizacím je poskytnuta ve 3D.	M	I
<b>A.C</b>		<b>Interaktivita</b>			
A.C	A.C.1	Propojení legenda – mapa	Legenda je interaktivní a umožňuje úkony s mapou	S	I
	A.C.2	Zvýraznění	Kliknutí na legendu zvýrazní prvek v mapě.	S	I
	A.C.3	Vypnout / Zapnout	Kliknutí na legendu zobrazí / schová prvek v mapě.	S	I

**A.A** – Zobrazení vyjmenovaných vrstev spočívalo pouze v nastavení přesné polohy, jelikož vrstvy připravené v Blenderu jsou skrze glTF formát převedeny do Wodnerland enigu v identické podobě.

**A.B** – Legenda byla manuálně modelována v Blenderu, tak aby schematicky znázornila příslušný tvar střechy, tedy doménu vstupního atributu STRECHA\_KOD.

**A.C** – Interakce byla implementována skrze interní kolizní systém, který Wonderland poskytuje. Pro dosažení zmíněné funkcionality byly napsány vlastní komponenty<sup>12</sup> (*toggle\_active*, *toggle\_highlight*). Tyto komponenty byly přiřazeny daným elementům legendy a umožnili zvýraznit, popř. vypnout / zapnout elementy v mapě.

<sup>12</sup> [Zdrojový kód.](#)

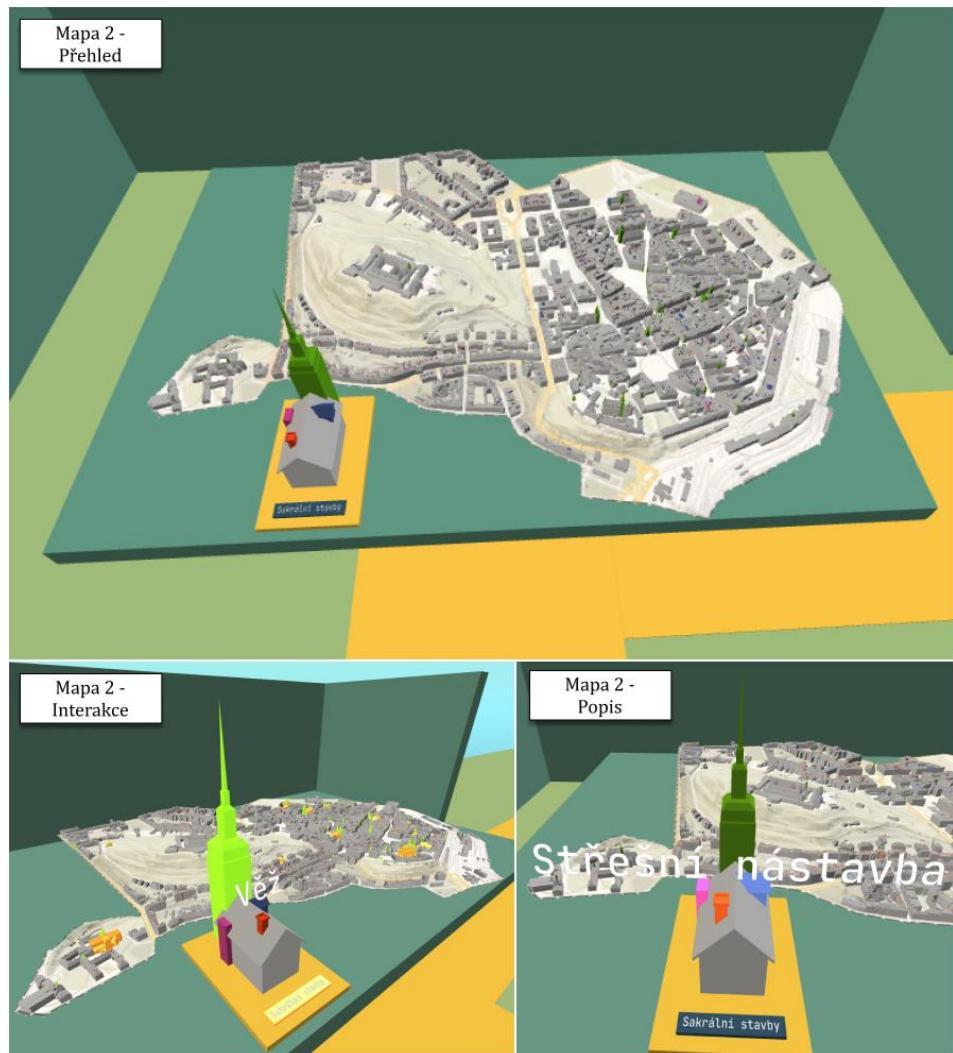


Obr. 30 Vizualizace č. 1

### Tematická mapa 2 – Části budov

Tab. 13 Matice požadavků – B – Tematická mapa 2 - Části budov

ID	Požadavek	Poznámka	Priorita	Status
B.A	<b>Vizualizace mapového pole</b>			
B.A	B.A.1	Zobrazení budov	Zobrazení vrstvy 3D budov.	M
	B.A.2	Zobrazení terénu	Zobrazení vrstvy 3D terénu.	M
	B.A.3	Texturování terénu	Zobrazení 3D terénu s texturou.	M
B.B	<b>Vizualizace legendy</b>			
	B.B.1	Legenda ve 3D	Legenda k vizualizacím je poskytnuta ve 3D.	M
B.C	<b>Interaktivita</b>			
B.C	B.C.1	Propojení legenda – mapa	Legenda je interaktivní a umožňuje úkony s mapou	M
	B.C.2	Zvýraznění	Kliknutí na legendu zvýrazní prvek v mapě.	M
	B.C.3	Popis	Namířením na prvek v legendě se zobrazí popis daného prvku.	S



Obr. 31 Vizualizace č. 2

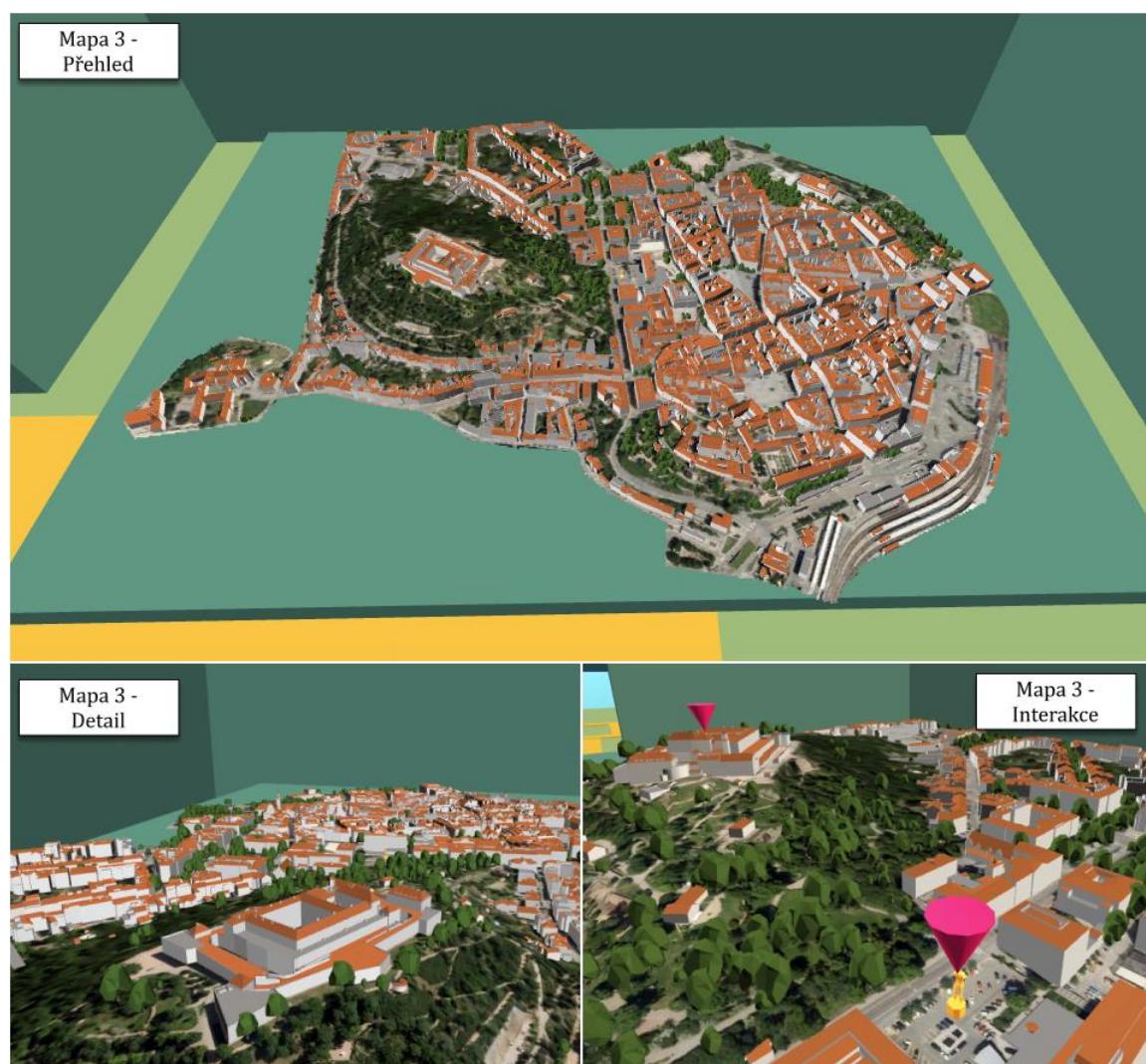
**B.B** – Obdobně jako u A.B byla vytvořena schematická 3D legenda znázorňující doménu atributu CAST\_OBJEKTU.

**B.C** – Interakce zde byla implementována obdobně jako u Mapy. č. 1 s tím rozdílem, že byla odebrána funkci vypínání / zapínání. Kliknutím uživateli zvýrazní budovu, namířením na příslušnou část legendy se uživateli zobrazí název dané kategorie.

## Topografická mapa

Tab. 14 Matice požadavků - C - Topografická mapa

ID	Požadavek	Poznámka	Priorita	Status
C.A	<b>Vizualizace mapového pole</b>			
C.A	C.A.1	Zobrazení budov	Zobrazení vrstvy 3D budov.	M I
	C.A.2	Zobrazení terénu	Zobrazení vrstvy 3D terénu.	M I
	C.A.3	Texturování terénu	Zobrazení 3D terénu s texturou.	M I
C.B	<b>Interaktivita</b>			
	C.B.1	Interaktivní objekty mapového pole	Zájmové objekty jsou zvýrazněny na základě kliknutí.	S I



Obr. 32 Vizualizace č. 3

**C.B** – V případě topografické mapy se interaktivita pojila k úkolu v rámci uživatelského testování, kdy byl uživatel požádán aby vyhledal 5 objektů (soch) v mapě. Sochy na kliknutí byly zvýrazněny kuželem.

## Mimo – funkční požadavky

Tab. 15 Mimo-funkční požadavky aplikace.

ID	Požadavek	Poznámka	Priorita	Status
E.1	Přístupnost	Aplikace by měla být dostupná prostřednictvím webového rozhraní.	M	I
E.2	Podpora	Aplikace by měla být podporována moderními webovými prohlížeči.	M	I
E.3	Dostupnost	Aplikace by měla být dostupná 99% času.	S	I
E.4	Otevřenost	Aplikace by měla být open-source a měla by využívat open-source technologie.	C	PI
E.5	Dokumentace	Aplikace by měla poskytovat příslušnou dokumentaci.	C	PI
E.6	Responzivita	Aplikace by měla být přístupná a použitelná z různých zařízení (HMD, mobil, desktop).	S	I
E.7	Výkon	Aplikace by na cílových zařízeních neměla vykazovat nižší průměrné hodnoty FPS nežli 50.	S	PI

**E.1, E.2** – Aplikace je publikována v prostředí Gitlab Pages. Jedná se o bezplatnou možnost publikace webové aplikace přímo z GitLab repositáře (viz. kap. 5.3.3 CI / CD). Aplikace je tedy dostupná z libovolného prohlížeče, který podporuje potřebné API.

**E.3** – Dostupnost aplikace je těžké změřit, toto kritérium je definováno stabilitou serverů platformy GitLab.

**E.4** – Tento požadavek nebyl naplněn zcela, jelikož Wonderland engine je *closed source* řešení. Zobrazovaná data jsou vytvořena skrze striktně proprietární licencovaný software (ArcGIS Pro, City Engine). Požadavek je možné považovat za částečně naplněný, jelikož Wonderland engine je možné využívat bez licence pro nekomerční účely.

**E.5** – Dokumentací aplikace je tento text.

**E.6** – Aplikace byla vyvinuta tak aby bylo možné její plné použití v HMD zařízení s 6DOF ovladači, na tradičním obrazovce s klávesnicí a myší a na mobilním zařízení. V HMD je uživatel schopen se pohybovat pomocí joysticků, popř. pohybu v lokálním prostoru. V případě desktopu je pohyb realizován pomocí kláves WSAD a kurzorem myši. Na mobilním zařízení je pohyb realizován pomocí virtuálních joysticků.

**E.7** – Aplikace zobrazuje velké množství dat a to jak geometrických (vyšší statisíce trojúhelníků) tak rastrových (3x textura o rozlišení 4096x4096) bylo nutné vynechat dynamické osvětlení (simulaci stínů) a minimalizovat počet světel na 2. Tvorba statického osvětlení pomocí zapékání byla otestována. Tuto metodu však nebylo možné použít, jelikož data budov nemají vhodnou geometrii. Výkon aplikace na hlavní cílové platformě (Oculus Quest 2) neklesl pod 40 FPS.

### 5.3.3 CI / CD

Jak bylo naznačeno v kap. Vývojářské nástroje, proces CI/CD hraje klíčovou roli při vývoji VR aplikace na webu. Tento proces umožňuje průběžnou integraci změn a automatické nasazování webové aplikace, která je následně dostupná prostřednictvím URL. V případě tradičních

webových aplikací, jako jsou HTML, JS a CSS, lze tento proces relativně jednoduše nastavit na platformách jako GitHub nebo GitLab, což usnadňuje testování a vývoj. Příklady takových aplikací mohou zahrnovat testovací aplikace pro Aframe, Three.js a podobně (viz. Tab. 8).

V kontextu Wonderland aplikace je proces složitější, neboť výsledná aplikace je binární soubor vytvořený pomocí Wonderland enginu. Z tohoto důvodu byly vytvořeny vlastní CI instrukce ve formátu YML, který obsahuje specifické kroky pro vykonání na zvolené CI/CD platformě. Tyto instrukce jsou implementovány v přiloženém skriptu (viz. Příloha č. 3). Proces funguje tak, že při každé změně obsahu vzdáleného repositáře Wonderland engine automaticky vystaví novou binární verzi aplikace a tuto verzi následně publikuje. Přiložený skript je specifický pro platformu GitLab, každopádně GitHub poskytuje obdobnou funkcionality skrze *GitHub Actions*.

## 6 UŽIVATELSKÉ TESTOVÁNÍ

Za účelem zhodnocení aplikace bylo provedeno uživatelské testování. Testování bylo zaměřeno primárně na použitelnost aplikace. Testování bylo inspirováno obecným přístupem k testování použitelnosti, a to dělením na testování správnosti / chybovosti (*effectiveness*), rychlosti vypracování (*efficiency*) a spokojenosti (*satisfaction*). Druhým dělením je dělení uživatelského testování na kvalitativní a kvantitativní přístup, kdy kvantitativní vychází z přesných měření a kvalitativní se snaží získat subjektivní informaci od testovaného uživatele. Autoři (Sterba et al. 2015) tvrdí, že v uživatelském testování kartografických výstupů je možná souhra výše zmíněných dělení.

Uživatelské testování nabízí řadu metod jako např. rozhovor, a/b testování, *eye-tracking*, *user logging* aj. Metodami je nutné měřit chování uživatele v rámci aplikace, popř. v případě kartografie jejich práce s mapou. Z tohoto důvodu je nutné, aby uživatel vykonal navržený praktický úkol. Zvolenými metodami je pak možné řešit správnost, efektivitu a spokojenosť. Za účelem testování konkrétní vizualizace / aplikace se zpravidla využívá více metod. Tento přístup se pak nazývá *Mixed research design*. Uživatelské testování v kartografii je souhou 3 hlavních komponent jimiž jsou Uživatelé, Vizualizace a Úkoly. (Šašinka 2013)

### 6.1 Návrh testování

Cílem testování bylo zjištění použitelnosti aplikace a samotných vizualizací. Pro každou z vizualizací byl přiřazen úkol, který uživatel splní. Návrh testování byl primárně ovlivněn možnostmi vyvinuté aplikace. Jelikož v rámci aplikace nebyla implementována funkciaálnita rozpoznaní vykonání úkolu a následné uložení výsledku, bylo nutné zvolit osobní asistované testování. Zároveň nebyla při testování možnost externího monitorování chování uživatele ve VP. Testování tedy spočívalo v uživatelově verbální interakci s koordinátorem, kdy uživatel komunikuje svoji aktivitu v rámci prostředí (zdali vypracovává úkol, přesouvá se mezi úkoly atd.). Testování proběhlo za použití HMD Oculus Quest 2 256 GB.

#### 6.1.1 Průběh testování

Průběh testování byl rozdělen do 3 částí.

1. **Příprava** – Proces seznámení se s HMD, vysvětlení způsobu interakce ve VP (způsob interakce s prostředím, pohyb), instrukce pro úspěšné spuštění aplikace.
2. **Průchod VP** – Vlastní průchod scénou za hlasové asistence koordinátora. Výsledky úkolů jsou zapisovány koordinátorem.
3. **Vyplnění dotazníku** – Uživatel vyplní dotazník.

V průběhu celého testování byl nahráván zvuk (interakce koordinátora a uživatele) a měřen čas vykonání jednotlivých částí Průchodu VP. Dotazník byl sestaven z 5 částí. Úvodní identifikační část obsahující otázky na charakteristiku uživatelů (věk, vzdělání atd.). Následně 3 sekce otázek na jednotlivé úkoly. Sekce zjišťující informace ohledně ovládání aplikace a poslední sekce s otevřenými otázkami pro získání osobního názoru respondenta. V dotazníku byly použity 3 typy otázek. Nejvíce bylo využito typu Likertovy škály. V menší míře pak byly použity otázky klasifikační a otázky otevřené, za účelem získání názoru uživatele. Dotazník byl realizován skrze platformu Google Forms. Přehled otázek z dotazníku je možné nalézt v Příloze č. 4. Cílovou skupinou testování byly zaměstnanci Kanceláře Architekta města Brno. Jednalo se převážně o odborníky na územní plánování a městský rozvoj. Otestováno bylo 11 osob.



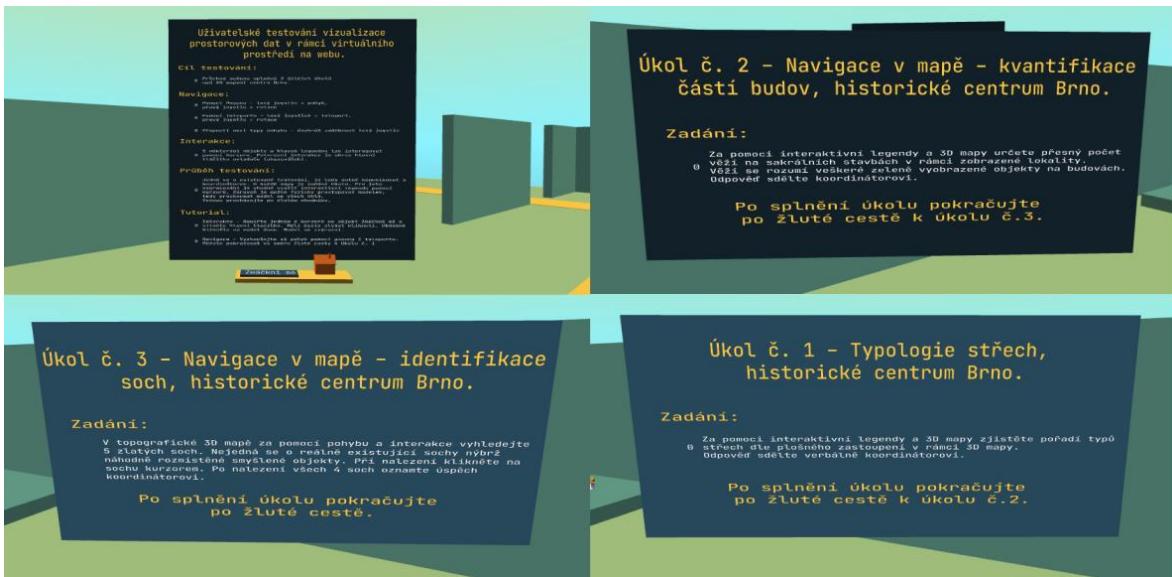
Obr. 33 Průběh uživatelského testování v kanceláři KAM.

### 6.1.2 Průchod VP

Testování probíhalo za asistence koordinátora. Každopádně aplikace byla navržena tak aby uživatel byl schopen samostatné navigace a plnění úkolů viz. Obr. 34. Byly zvoleny 3 typy úkolů na základě taxonomie uvedené v (Sterba et al. 2015). Vyhotovení úkolů bylo zapisováno koordinátorem. Při vstupu do VP byl respondent požádán o vypracování cvičné úlohy, kdy byl textově vysvětlen průběh testování, navigace a především interakce. Interakce s objekty byla prezentována pomocí exemplární interaktivní legendy, na jejíž objekty mohl uživatel „kliknout“. Ovládání pohybu bylo následně uživateli vysvětleno v rámci cvičné úlohy.

Tab. 16 Zadání úkolů.

ID	Zadání	Typ
Úkol. 1	Za pomoci interaktivní legendy a 3D mapy zjistěte pořadí typů střech dle plošného zastoupení v rámci 3D mapy.	Porovnání, Seřazení
Úkol. 2	Za pomoci interaktivní legendy a 3D mapy určete přesný počet věží na sakrálních stavbách v rámci zobrazené lokality. Věží se rozumí veškeré zeleně vyobrazené objekty na budovách.	Identifikace, Součet
Úkol. 3	V topografické 3D mapě za pomocí pohybu a interakce vyhledejte 5 zlatých soch. Nejedná se o reálně existující sochy nýbrž náhodně rozmištěné smyšlené objekty. Při nalezení klikněte na sochu kurzorem.	Identifikace, Lokace

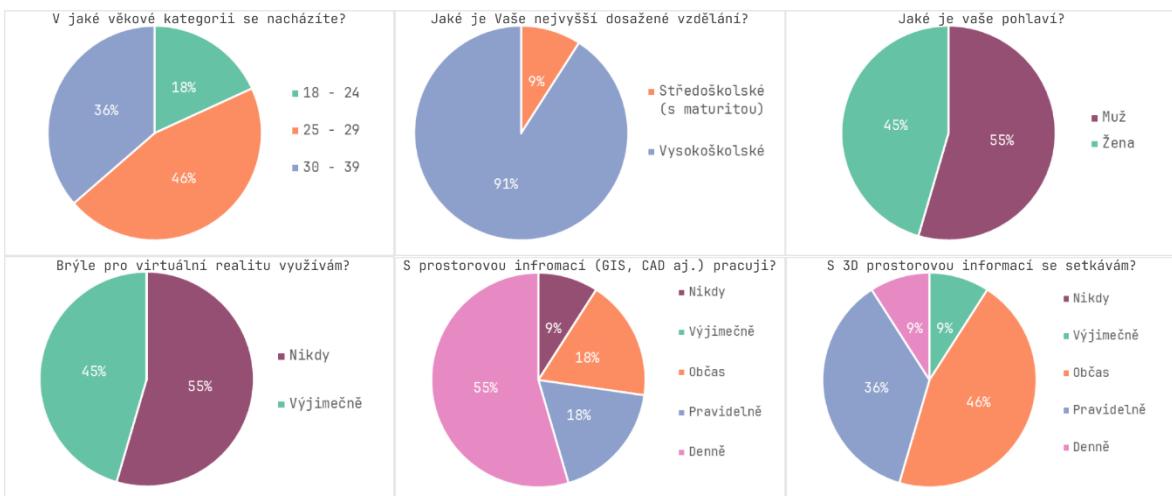


Obr. 34 Instrukční textové tabule se zadáním úkolů.

## 6.2 Výsledky

Výsledky uživatelského testování byly vyplňený dotazník a dosažené skóre v rámci vypracování úkolů. Pro úkoly č. 1 a 2 byl definován bodovací systém. Jelikož se jednalo o rozdílné úkoly bodovací systém byl rozdílný, není tedy možné přímo porovnat dosažené hodnoty, lze pouze porovnat úspěšnost daných účastníků mezi sebou v rámci jednoho úkolu. V případě úkolu č. 1 se jednalo o řazení kategorií dle četnosti v mapě. Body byly přiřazeny podle procentuální úspěšnosti. Úkol č. 2 spočíval v určení přesného počtu definovaných objektů. Výsledné skóre pro úkol č. 2 bylo vypočteno tak, že byla hodnocena odchylka od správné odpovědi. Jelikož rozptyl odpovědí byl značný byl výpočet vážen tak aby zahrnul celý vzorek odpovědí. U úkolu č. 3 byl hodnocen pouze čas, jelikož se jednalo o nalezení objektů v mapě. Pro výsledky úspěšnosti úkolů byla zpracována základní deskriptivní statistika. Výsledky dotazníkového šetření byly analyzovány graficky. Podrobnější statistické zpracování nebylo provedeno z hlediska, že se jedná o nízký počet respondentů.

### 6.2.1 Identifikace



Obr. 35 Výsledky dotazníkového šetření – sekce Identifikace (otázky 1-7)

Výstupy dotazníkového šetření byly rozděleny dle sekcí v dotazníků. Na základě výsledků z identifikační sekce je možné pozorovat, že testovaná skupina byla primárně vysokoškolského vzdělání. Věkové zastoupení je převážně mezi 25–39 lety. Z hlediska predispozice vůči technologii a zaměření vizualizací výsledky ukazují, že uživatelé nejsou běžnými uživateli HMD, tudíž ovládání a obecná práce s touto technologií je pro ně nová. Jelikož zájmovou skupinou byli zaměstnanci KAM, většina má denní, popř. velmi častou interakci s prostorovými daty. V případě 3D dat je zkušenost méně extrémní, ačkoliv stále vysoká. Právě predispozice práce s prostorovými daty a zkušeností s 3D je pravděpodobně značně vyšší nežli v obecné populaci. Z hlediska hodnocení dalších výstupů to není nevhodné, jelikož samotná aplikace míří na specialisty více nežli na obecnou populaci.

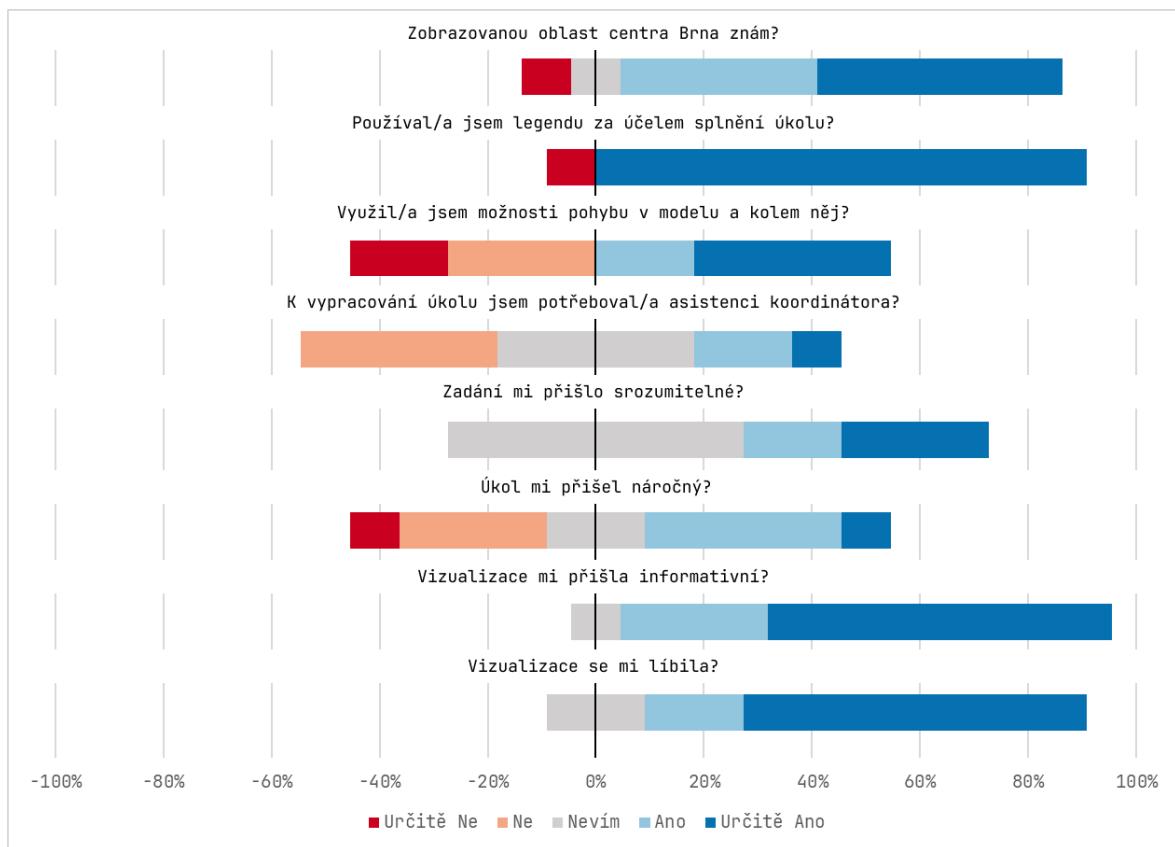
### 6.2.2 Úkoly

V případě úkolu č. 1 je zahrnuta otázka o znalosti území. Zde většina respondentů zná lokalitu velmi dobře. Je však možné tvrdit, že vzhledem k typu úkolů nemá znalost oblasti vliv na správnost vypracování.

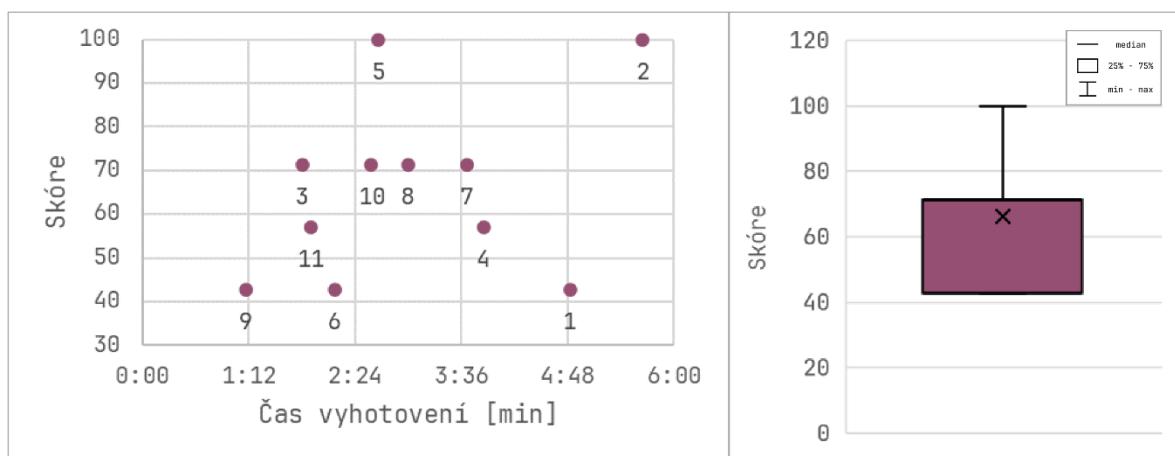
Při analýze otázek zaměřujících se na interakci s prostředím výsledky ukazují na to, že většina respondentů využila jak interakce s objekty, tak pohybu (interaktivní legenda, pohyb) při vypracování úkolů. Při porovnání mezi úkoly je patrný vzrůst míry interakce pomocí pohybu U2 a U3. Tento fakt je možné připsat charakteru úkolů, který tuto interakci přímo vyžadoval, ale také skutečnosti, že uživatel se lépe naučil ovládání pohybu ve VP na rozdíl od U1. Což může naznačovat potřebu delší cvičné úlohy, kdy si uživatel osvojí ovládání aplikace.

Při analýze otázek na způsob zadání úkolů a potřebu asistence je možné u U1 a U2 pozorovat skutečnost, že výrazná část respondentů vyžadovala asistenci koordinátora, ačkoliv odpovědi na srozumitelnost zadání jsou převážně neutrální až pozitivní, tedy že respondentům přišla zadání srozumitelná. Zde i na základě analýzy zvukových záznamů z testování je možné tvrdit, že v případě U1 a ve větší míře u U2 byla zadání interpretována jinak, než bylo zamýšleno v návrhu. V případě U1 byla nutnost častého dovytváření zadání.

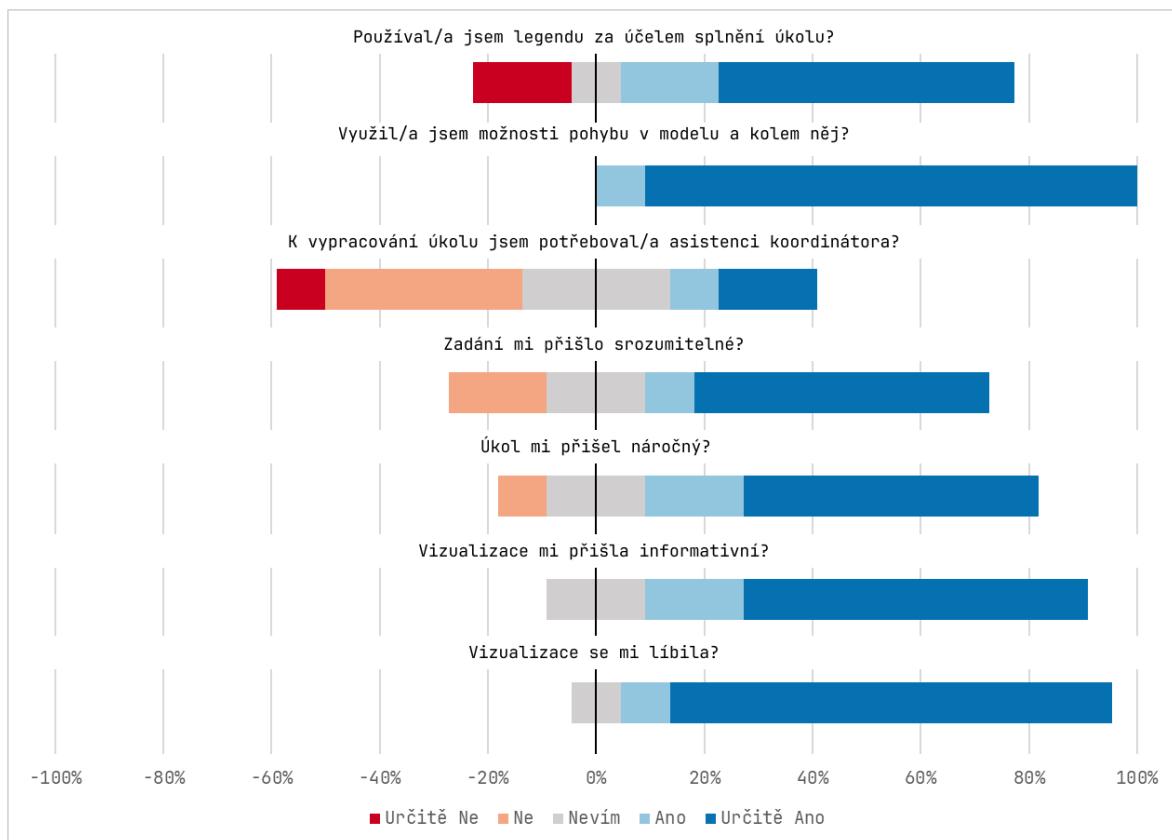
Otzáka na zdánlivou náročnost úkolů ukázala, že U2 přišel uživatelům nejsložitější druhý v pořadí byl U1 a nejjednodušším byl U3. Tuto skutečnost je možné pozorovat také na krabicovém grafu Obr. 41, který ukazuje čas vyhotovení napříč respondenty. Složitost úkolů plyne především z typu úkolu. U2 vyžadoval počítání objektů ( $n = 33$ ) v 3D modelu, což se ukázalo jako kognitivně náročná činnost v kombinaci s orientací v 3D prostoru. Běžným průběhem vypracování U2 byla situace kdy respondent opakovaně počítal objekty, jelikož při pohybu v 3D prostoru ztratil pojem o aktuálním počtu, popř. které již spočítal a které ne. Zde je nutné podotknout částečnou nevhodnost úkolu, jelikož manuální počítání by v reálném případě mělo být nahrazeno funkcionálitou (analytickým nástrojem), která uživateli požadované objekty spočítá, tedy zvýší rychlosť vypracování. Z pohledu spokojenosti s danými úkoly jsou odpovědi neutrální až převážně pozitivního sentimentu.



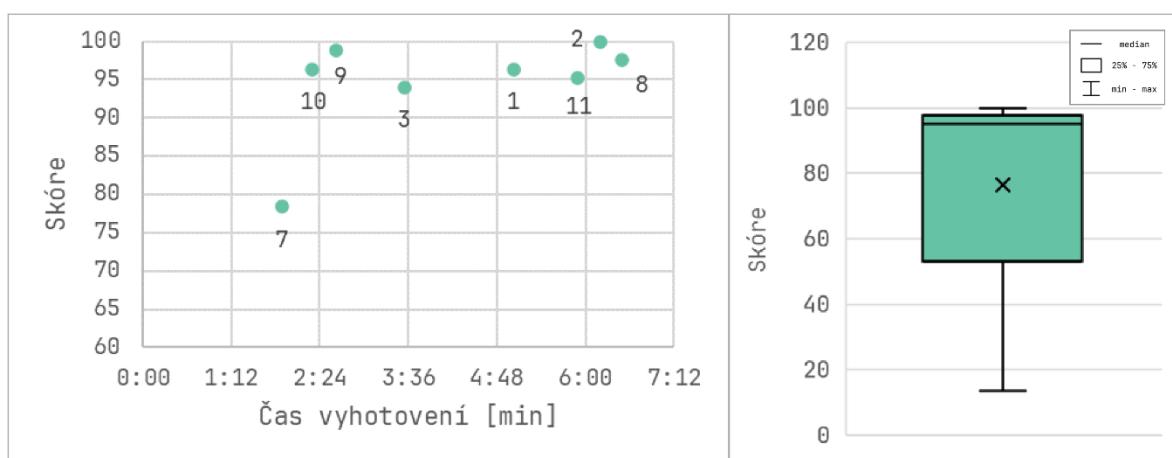
Obr. 36 Výsledky dotazníkového šetření – sekce Úkol č. 1 (otázky 7-14)



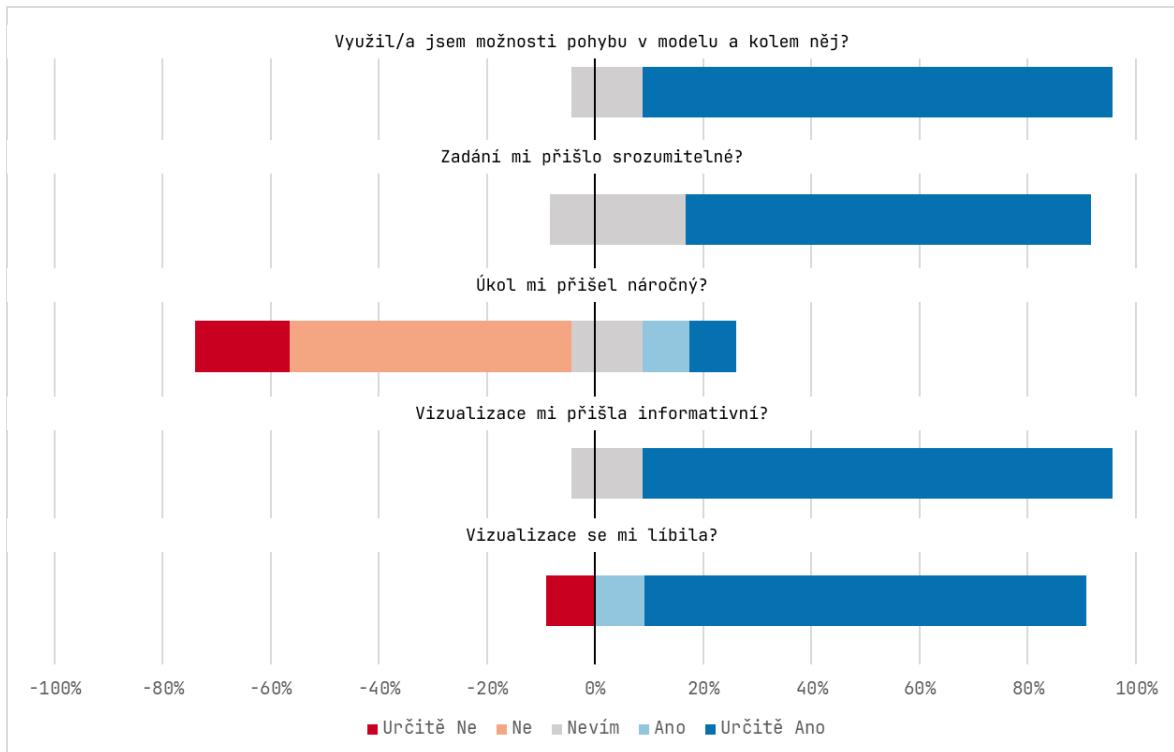
Obr. 37 Správnost vyhotovení úkolu č. 1 v závislosti na čase – vlevo. Deskriptivní statistika výsledků – vpravo.



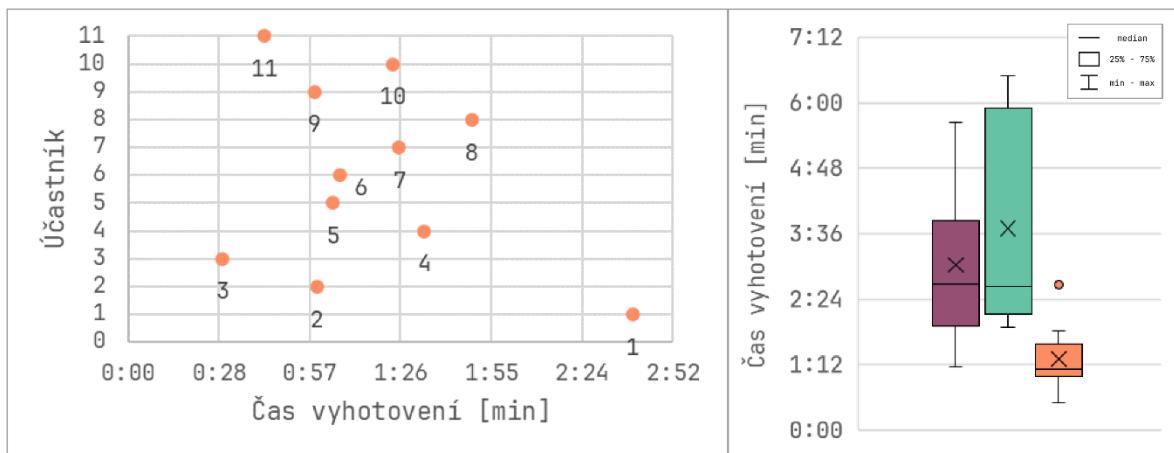
Obr. 38 Výsledky dotazníkového šetření – sekce Úkol č. 2 (otázky 15-21)



Obr. 39 Správnost vyhotovení úkolu č. 2 v závislosti na čase – vlevo. Deskriptivní statistika výsledků – vpravo.



Obr. 40 Výsledky dotazníkového šetření – sekce Úkol č. 3 (otázky 22-26)

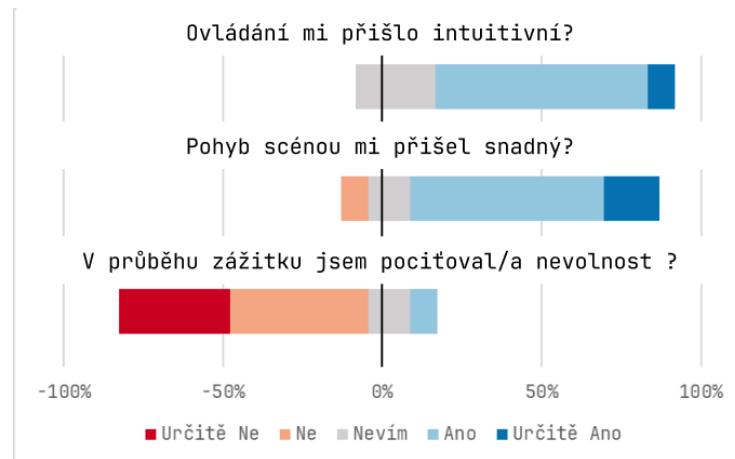


Obr. 41 Rychlosť vyhotovení úkolu č. 3 – vlevo. Deskriptívne statistiky časov vyhotovení úkolu, barevné rozdelené podľa úkolu – vpravo.

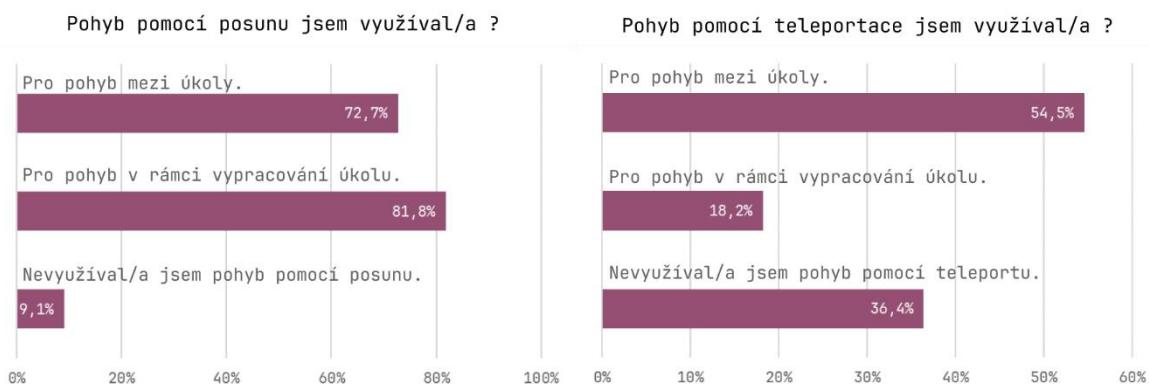
### 6.2.3 Ovládání

Výsledky uživatelského testovania sekcie ovládania vykazujú prevažný sentiment v prospèch návrhu aplikacie. Z Obr. 52 je patrné, že většinu uživatelů přišlo ovládání intuitívne a pohyb snadný. Jelikož většina respondentů měla před testováním minimální zkušenosť s virtuálnimi brýlemi je možné tyto výsledky považovať za ukazatele vhodného návrhu interakcie. Zároveň mezi respondenty nebyl případ nevolnosti, kdy by bylo nutné přerušit testování. Při interpretaci absence nevolnosti je nutné vzít v potaz, že se jednalo o krátké testování s průměrným časem 15 minut. Je možné spekulovat, že při delším užívání by míra nevolnosti vzrostla. Uživatelé využívali prevažně pohyb pomocí posunu, a to jak pro pohyb mezi úkoly, tak pro pohyb při vypracovávání úkolu. Velký vliv na tyto metriky měla skutečnosť, že pohyb

pomocí posunu byl výchozím nastavení pro pohyb a pro jeho změnu bylo nutné dvakrát stlačit tlačítko na ovladači.



Obr. 42 Výsledky dotazníkového šetření – sekce Ovládání (otázky 27, 28, 31)



Obr. 43 Výsledky dotazníkového šetření – sekce Ovládání (otázky 29, 30)

#### 6.2.4 Osobní názor

V poslední sekci dotazníku „Otevřené otázky“ (otázky č. 33, 34) byl analyzován osobní názor na VP obecně.

- Jak obecně vidíte potenciál VR v praxi?
- Pokud máte jakékoli další poznámky či komentáře tak uved'te. (nepovinné)

V řadě odpovědí se opakovala stejná témata. Většina účastníků vyjádřila mírně až výrazně pozitivní sentiment k vývoji využití VR v praxi a budoucnosti vývoje této technologie, popř. její adopce do běžného života. Nejvíce zmiňovanými tématy uplatnění VR bylo téma urbanismu, a to především v kontextu tvorby vizualizací pro komunikaci koncepcí území, popř. architektonických projektů. Vícekrát byla zmíněna využitelnost VR v případě, že aplikace umožňuje kolaborativní prostředí.

Mezi zmíněnými negativy se opakovaně objevovala nespokojenosť se samotným zařízením. Opakovaně se objevily připomínky k nepohodlí způsobené virtuální helmou. Dále bylo zmiňováno nedostatečné rozlišení displejů. V případě účastníků s nedokonalým zrakem byl a použitelnost aplikace značně snížena. V případě extrémní krátkozrakosti byla aplikace bez asistence koordinátora nepoužitelná, jelikož nebylo možné přečíst instrukce v rámci aplikace.

## 7 DISKUZE

### Osnova

#### Zhodnocení dosažených výsledků:

- Představený přehled technologií považuji za dostačující.
- Funkční VP aplikace implementována.
- Aplikace otestována
- Podrobně popsán process transformace dat a tvorby VP – publikovány vyvinuté nástroje

#### Srovnání s existující literaturou a rešerší:

- Porovnání vašich výsledků s existujícími teoretickými základy a předchozími výzkumy.
  - (Laksono, Aditya 2019), (Rzeszewski, Orylski 2021)
- Project FlowerBed, ParAdowski projekty, VS moje implmentace
  - o Týmové projekty, I dlouhodobé, výsledkem často poměrně jednoduchá aplikace.
- Geoprostorová řešení VS moje apka

#### Vliv rozhodnutí a metod na výsledky, kritika rozhodnutí:

- **Datový pohled**
  - o Má cenu dávat data do VR
  - o Datový technický dluh – data jsou vytvořena bez myšlenky jejich využití – vede k nutnosti zbytečných úprav, je možné předejít následováním standardních postupů
  - o Dynamická data dynamické měřítko – Budoucnost geodat ve VR a ve 3D je v instancovaných datech. - <https://github.com/pka/awesome-3d-tiles>
  - o Propojení datových modelů – GIS = atributy + geometrie, 3D viz = material + geometrie – glTF geospatial profile – potřeba exportérů a nástrojů pro převod z GIS formátů
- **Vývojový pohled**
  - o Optimalizace – stále příliš nutná – mělo by být automatizováno
  - o Ani není potřeba tolik „programování“ vyvinout VR aplikaci je možné pouze „klikáním“
- **Uživatelské testování**
  - o upravit aplikaci tak aby testování mohlo být provedeno unsupervised a mohla být otestována usabilita aplikace a tím samotné technologie namísto specifických úkolů
  - o Lépe specifikovat úkoly – KISS keep it simple stupid – navrh ukolu není realny, počítat veze by měl gis ne člověk
  - o Detailnější dotazník – identifikovat interakci a pohyb.
  - o User logging – má interakce vliv na správnost vyhotovení?

- Vyhodnocení – lepší by bylo použít SUS namísto subjektivního vyhodnocení výsledků

#### **Budoucnost, praktický vliv?**

- Když bude lehke prevadet data z GIS do 3D tak vývoj VR už je na dost dobré úrovni
- WebXR a obecně VR je pořád hodně brzo – technologie se rapidně vyvíjí

## **8 ZÁVĚR**

Osnova:

**Shrnutí významu**

**Shrnutí výsledků**

**Shrnutí co dál?**

**Shrnutí k čemu to bylo**

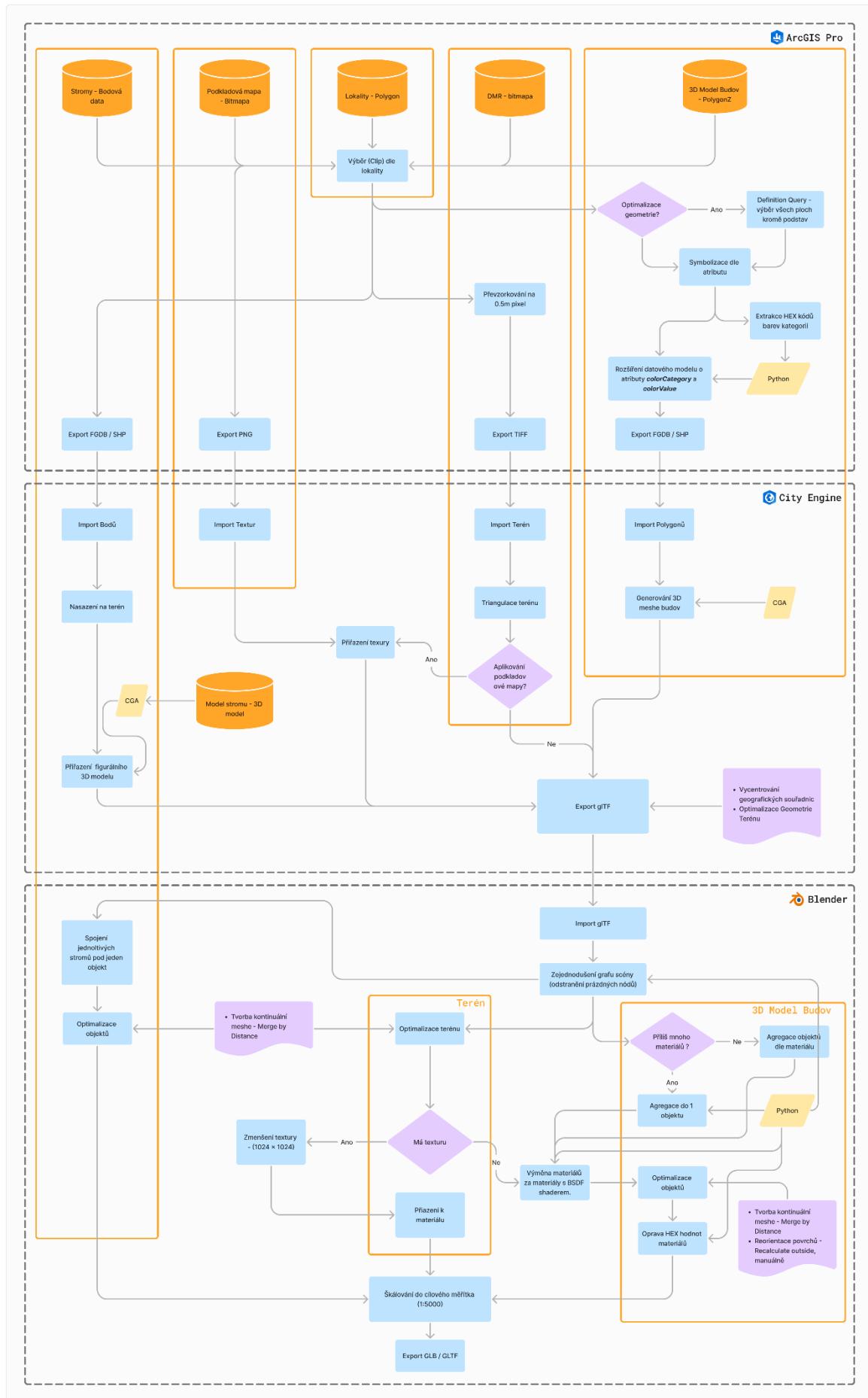
## **BIBLIOGRAFIE**

## **PŘÍLOHY**

## **SEZNAM PŘÍLOH**

Příloha č

Nástroj		Počet vertexů				Disk [MB]		GPU [MB]	
Merge By Distance	Primitiva	Vstup	Výstup	Δ	Δ [%]	Vstup	Výstup	Vstup	Výstup
Budovy	Vertex	461,694	91,278	370,416	80.23				
	Hrana	461,694	248,327	213,367	46.21				
	Plocha	153,898	153,898	0	0.00				
Terén	Vertex	580,602	98,777	481,825	82.99				
	Hrana	580,602	292,310	288,292	49.65				
	Plocha	193,534	193,534	0	0.00				
Stromy	Vertex	494,700	87,426	407,274	82.33				
	Hrana	494,700	242,850	251,850	50.91	93.8	27.6	843.1	30.3
	Plocha	164,900	161,900	3,000	1.82				
Decimate - 0.3	Primitiva	Vstup	Výstup	Δ	Δ [%]				
Terén	Vertex	98,777	20,492	78,285	79.25				
	Hrana	292,310	59,196	233,114	79.75				
	Plocha	193,534	38,705	154,829	80.00				
Resize texture	Rozlišení - px	14453 x 10624	1024 x 1024	/	/				



```
1. stages:
2.   - build
3.
4.   - deploy
5.
6. package:
7.   image: wonderlandengine/editor:1.1.4
8.
9.   stage: build
10.
11. script:
12.     # Set up the WLE_CREDENTIALS variable for WonderlandEditor to log in
13.
14.     - WonderlandEditor --windowless --package --project wle-pplayground-maps.wlp
15.
16. cache:
17.   key: ${CI_COMMIT_REF_SLUG}
18.
19.   paths:
20.     - cache/
21.
22. artifacts:
23.   paths:
24.     - deploy
25.
26. pages:
27.   image: alpine:3.14
28.
29.   stage: deploy
30.
31. rules:
32.   # Only deploy to pages on main/master branch
33.
34.   - if: $CI_COMMIT_BRANCH == $CI_DEFAULT_BRANCH
35.
36. before_script:
37.   - apk add gzip
38.
39. script:
40.   - gzip -k deploy/*
41.
42.   - mv deploy public
43.
44. artifacts:
45.   paths:
46.     - public
```

Příloha č. 4 Seznam otázek uživatelského testování.

<b>Typ otázek</b>	<b>ID</b>	<b>Otázka</b>
<b>Identifikační</b>	1	Jaké je vaše pohlaví?
	2	V jaké věkové kategorii se nacházíte?
	3	Jaké je Vaše nejvyšší dosažené vzdělání?
	4	Brýle pro virtuální realitu využívám?
	5	S 3D prostorovou informací se setkávám?
	6	S prostorovou informací (GIS, CAD aj.) pracuji?
	7	Zobrazovanou oblast centra Brna znám?
<b>Viz. 1</b>	8	Používal/a jsem legendu za účelem splnění úkolu?
	9	Využil/a jsem možnosti pohybu v modelu a kolem něj?
	10	K vypracování úkolu jsem potřeboval/a asistenci koordinátora?
	11	Zadání mi přišlo srozumitelné?
	12	Úkol mi přišel náročný?
	13	Vizualizace mi přišla informativní?
	14	Vizualizace se mi líbila?
<b>Viz. 2</b>	15	Používal/a jsem legendu za účelem splnění úkolu?
	16	Využil/a jsem možnosti pohybu v modelu a kolem něj?
	17	K vypracování úkolu jsem potřeboval/a asistenci koordinátora?
	18	Zadání mi přišlo srozumitelné?
	19	Úkol mi přišel náročný?
	20	Vizualizace mi přišla informativní?
	21	Vizualizace se mi líbila?
<b>Viz. 3</b>	22	Využil/a jsem možnosti pohybu v modelu a kolem něj?
	23	Zadání mi přišlo srozumitelné?
	24	Úkol mi přišel náročný?
	25	Vizualizace mi přišla informativní?
	26	Vizualizace se mi líbila?
<b>Ovládání</b>	27	Ovládání mi přišlo intuitivní?
	28	Pohyb scénou mi přišel snadný?
	29	Pohyb pomocí posunu jsem využíval/a?
	30	Pohyb pomocí teleportace jsem využíval/a?
	31	V průběhu zážitku jsem pocítoval/a nevolnost?
<b>Otevřené otázky</b>	32	Jak obecně vidíte potenciál VR v praxi?
	33	Pokud máte jakékoli další poznámky či komentáře tak uved'te. (nepovinné)