

MASARYKOVA  
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

**Autor/Autorka:** Bc. Jan Horák  
Přírodovědecká fakulta, Masarykova univerzita  
Geografický ústav

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

**Studijní program:** Geografická kartografie a geoinformatika

**Studijní obor:** Geografická kartografie a geoinformatika

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

**Akademický rok:** 2022/2023

**Počet stran:** xx

**Klíčová slova:** Klíčové slovo, Klíčové slovo, Klíčové slovo, Klíčové slovo,  
Klíčové slovo, Klíčové slovo, Klíčové slovo, Klíčové slovo

## BIBLIOGRAFICKÝ ENTRY

**Author:** Bc. Jan Horák  
Faculty of Science, Masaryk University  
Department of Geography

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

**Degree Programme:** Cartography and geoinformatics

**Field of Study:** Cartography and geoinformatics

**Supervisor:** RNDr. Lukáš Herman, Ph.D.

**Academic Year:** 2022/2023

**Number of Pages:** xx

**Keywords:** Keyword, Keyword, Keyword, Keyword, Keyword, Keyword, Keyword, Keyword

## ABSTRAKT

Tato práce se zabývá..

## ABSTRACT

Bachelor thesis..

M U N I  
S C I

Masarykova univerzita  
Přírodovědecká fakulta

## ZADÁNÍ BAKALÁŘSKÉ 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 funkcionality se může v různých podmínkách lišit. Na základě srovnání dostupných technologií bude vybrána technologie pro vytvoření vlastní vizualizace prostorových dat z vybrané aplikační oblasti. Vytvořená vizualizace bude následně uživatelsky evaluována a zhodnocena tak její funkcionality.

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: ?

Rozsah průvodní zprávy: ?

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.

<i>Jazyk závěrečné práce:</i>	čeština
<i>Vedoucí bakalářské práce:</i>	RNDr. Lukáš Herman, Ph.D.
<i>Datum zadání diplomové práce:</i>	?
<i>Datum odevzdání bakalářské práce:</i>	podle harmonogramu

RNDr. Vladimír Herber, CSc.  
pedagogický zástupce ředitele ústavu

**PODĚKOVÁNÍ**

Lukáš Herman, ... atd.

**PROHLÁŠENÍ (DODELAT ODSAZENÍ PODLE DÉLKY PODĚKOVÁNÍ)**

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

Brno ... 2023

Jan Horák

.....  
Jméno Příjmení

## **OBSAH**

--- obsah až bude ---



## 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 nebo 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 v tom digitálním s přístupem umožňujícím pocit přítomnosti blížící se reálnému zážitku. 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 jednak dostupnost tak i možnost sdílení, interakce, popř. vzájemné kolaborace což odborná literatura považuje za stěžejní pilíře virtuální reality.

Přestože existuje nabídka desktopových softwarů, které umožňují vytváření virtuálních realit, nejsou vždy snadno dostupné pro běžné uživatele. Mnoho lidí má zájem o atraktivní výsledky, které jim umožní vizualizovat svět kolem nich, avšak nechtějí naučit, popř. ani nejsou schopni pracovat s často složitými desktopovými programy. Právě v této souvislosti umožňují nástroje ve webovém prostředí alternativu. Volba webového prostředí jakožto platformy pro virtuální realitu však nepřináší pouze benefity. Ačkoliv odstraňuje mnohé překážky pro uživatele, v některých aspektech (detail, výkonost, komplexita) je přenáší na vývojáře / tvůrce virtuální reality. Webové prostředí samo o sobě je velice rozmanitý ekosystém technologií, k tomu navíc je nutné uvažovat možnosti hardwarových zobrazovacích a vstupních zařízení. Je tedy žádoucí vyjasnit, jak se v tomto systému zorientovat pro účel vizualizace geografických dat ve virtuální realitě.

Dosavadní způsob, jakým byla geografická informace vizualizována a komunikována se převážně soustředil na 2D reprezentaci, tedy na mapy. Avšak v dnešní digitální éře technologický vývoj vede k tomu, že je možné si představit a prozkoumávat prostor ve třetí dimenzi. Zde se naskytá otázka, jaký vztah existuje mezi tradičními mapami a novou formou virtuální reality? Je virtuální realita pouze dalším médium nebo znamená pokrok ve způsobu, jakým komunikujeme prostorové informace? Je možné virtuální realitu zobrazující geoprostorovou informaci považovat za interaktivní 3D mapu? Kde leží hranice mezi mapou a virtuální realitou do jaké míry jsou oba tyto prostředky abstrakcí reality? Tato práce na tyto otázky nezodpovídá, ale snaží se dát kontext pro jejich zodpovězení ve formě znalosti možností nástrojů umožňujících tvorbu virtuální reality na webu.

Práce se nejdříve zaměří na teoretické základy virtuální reality. Následně na analýzu možných využití virtuální reality jakožto média pro vizualizaci geoprostorové informace. Klíčovou částí pak bude analýza soudobých technologií a jejich zhodnocení spolu s pilotní implementací webové aplikace zobrazující geografická data **pro vybraný účel**. Finální částí pak bude uživatelské zhodnocení výsledné pilotní aplikace.

## **1.1 Cíle práce**

Hlavní cíle práce:

- Analýza a zhodnocení aktuálních 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) a jejich propojení, umožňujících tvorbu virtuálních prostředí s geografickou konotací v rámci webového prostředí. Mimo výslednou vizualizační aplikaci by výstupem práce bude i odborné doporučení, jaké technologie jsou vhodné s ohledem na zvolený účel.

Dalším cílem této práce je vytvoření a následná uživatelská evaluace funkcionality webové aplikace využívající prostředků virtuální reality, jakožto nástroje k prezentaci geografických dat. Aplikace bude vytvořena na základě analýzy technologií v první fázi práce. Tato praktická část bude následně podkladem pro odborné doporučení vhodnosti technologií pro daný účel.

### **1.1.1 Výzkumné otázky**

**#TODO – jak správně formulovat**

Jsou současné technologie umožňující tvorbu virtuální reality ve webovém prostředí vhodné pro prezentaci geografických dat?

Jaké jsou faktory ovlivňující volbu technologie pro vizualizaci dat ve VR.

## 2 METODIKA

1. Zpracovat metodiku VR, 3D modelování.
2. Popsat základní technologie umožňující VR na webu – SW – základní API WebGL, XR atd.
3. Seznam a popsat nástavby nad základními API
4. Vybrat tech
5. Udělat appku 😊

V první části práce byla zpracována teoretická

Vlastní vizualizace bude vytvořena na základě podrobného průzkumu technologií v kontextu geografické vizualizace. Není možné vytvořit funkční mapu / vizualizaci / aplikaci, bez důkladného promyšlení příčin, které předurčují splnění účelu dané práce (Sterba et al. 2015). Z tohoto důvodu je nutné identifikovat geoprostorová témata pro která je vhodné využít vizualizace v rámci virtuální reality. Na základě těchto témat je nutné určit, jaká využívají geografická data a jaké dopady mají tato data na volbu postupů a technologií při jejich vizualizaci. Právě tyto prerekvizity a další specifické jako je aktuálnost, standardizace otevřenost aj. je nutné mít na paměti při analýze a následném výběru technologií pro vizualizaci. Z tohoto důvodu je nutné vytvořit sadu požadavků.

### 3 REŠERŠE – SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY

Za účelem získání obecného přehledu o problematice je vhodný průzkum obecných publikací jednak 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 v rámci virtuálních prostředí, je podrobné porozumění vstupním datům tedy datovým modelům, metodám zpracování a výměny mezi technologiemi (Keil et al. 2021). Jelikož předností virtuální reality je prezentace více rozměrných dat (šířka, délka, výška, popř. jiná veličina). Základy modelování takovýchto 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). řeší vývoj webového informačního systému pro publikaci 2D a 3D dat. 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). **#TODO Mozilla Hubs aj.** Vizualizaci velkého množství 3D budov (veškeré budovy v Holandsku) pomocí webového prohlížeče řeší.

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 využití 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 prostředích 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

**Commented [LH1]:** Tuto větu by chtělo nějak lépe navázat na předchozí. Například, že některé studie se soustředí na technické aspekty, například Cibula (2021)....  
Pozor taky na formátování citací - vím, že tohle je asi automaticky generované - mělo by být: Cibula (2021)

**Commented [JH2R1]:** .

se využitím 3D vizualizace zabývá (Judge, Harrie 2020). 3D webové vizualizace seismických hazardů infrastruktury využívají (Mazzei, Quaroni 2022). #TODO

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 geovizualizaci dat v rámci virtuální reality představuje (Bernardes et al. 2018). Geovizualizace v rámci tohoto výzkumu spočívala ve vytvoření virtuálních prostředí univerzitního kampusu pomocí herních enginů (např. Unity, Unreal).

Důležitým aspektem vývoje pro webové prostředí je porozumění ekosystému technologií, které jej umožňují. #todo – Obecně info o tom jak funguje web (scope zjistit ze studentských prací). Vývoj aplikací pro web zahrnující propojení s geoprostorovou informací se zabývá (Peňák 2017)

#TODO – rozdělit 3D a XR, dočíst víc článků kde: Web, XR a testování

#TODO Tematická kartografie?

#TODO Usability and user testing

## 4 TEORETICKÉ ZÁKLADY

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

#TODO – webový vývoj? – výhody a nevýhody zaměření na web – platformní nezávislost atd.  
22

### 4.24.1 Virtuální realita

Definovat koncept virtuální reality je obtížný úkol, převážně z důvodu, že se jedná o široký a z pohledu specifických technologií rychle měnící se pojem. Definice je navíc rozdílně interpretována napříč obory, které se jí 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)*

V oblasti kartografie a českém prostředí je možné se setkat s definicí pak virtuální realitu popř. virtualitu v návaznosti na virtuální geografická prostředí (VGE) definuje MacEachren pomocí 4 I virtuální reality (MacEachren et al. 1999) a to:

1. Imerze
2. Interaktivita
3. Informační intenzita
4. Inteligence objektů

Definice Shermana a Craiga pohlížející na virtuální realitu více z pohledu zážitku stejně jako LaValle, 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** a (stav zapojení do virtuální reality) **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 na základě zvoleného média (technologie) pomocí které je virtuální realita vytvořena. Výše zmíněné koncepty je možné přímo překrýt s MacEachrenovy, kromě konceptu **informační intenzity** a **inteligence objektů**, a to především protože 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 dále LOD a LOA**). Inteligencí objektů se pak chápá 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 zužívanou 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 kontinuu. XR je považován za novější termín, který zaštiťuje ostatní 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 fikční (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. Základním dělením virtuální reality podle míry imerze jsou 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)

Kdy 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 kombinace práce s reálným a virtuálním světem. (Kvarda 2020; Burdea, Coiffet 2003) Záměrem této práce je primárně imerzní VR tedy systém zobrazující virtuální svět v zobrazovacím zařízení (HMD). V rámci imerzního 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. Tato schopnost je primárně ovlivňována kompetencí hardwaru tudíž je možné jej dělit dle použitých zobrazovacích zařízení (vizuální, zvykové, haptické). Terminologie následně obsahuje tedy termíny jakožto mobile VR, desktop VR a HMD – immersive VR (viz. kap. Zobrazovací zařízení).

Formatted: Czech, Highlight

#### 4.2.14.1.1 Komunikační medium

Virtuální realita je médium, tudíž je možné ji chápat jako formu mezilidské komunikace a dále zkoumat její vztah k dalším formám komunikace (Sherman, Craig 2019). V rámci 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, kdy tvůrce je „odesílatelem“ informace a čtenář příjemce (Kubíček, Stachoň 2009). Sherman a Craig tuto myšlenku představují obecněji na různých formách médií. Společný je však fakt, že se jedná o lineární proces (Koláčný 1969), kdy tvůrce myšlenky si vybere médium (film, mapa, virtuální realita atd. ), následně vytvoří virtuální svět (příběh, reprezentace prostoru na mapě, reprezentace prostoru v počítači), který nás ledě realizuje pomocí technologií příslušící danému médiu, čímž vytváří pro účastníka jedinečný zážitek (Sherman, Craig 2019).



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

Výše zmíněný model, však není možné zcela aplikovat v případě přidání konceptu interaktivity 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.

#### 4.2.14.1.2 VR a kartografie

Kartografie v posledních 30 letech prošla tzv. „geoinformatizací“, kdy se od papírových map přešlo k digitalizaci a automatizaci analytických i vizualizačních procesů. Digitalizace dále umožnila nové způsoby interakce a dynamické práce s prostorovými daty, a to skrze geografické informační systémy GIS. (Konečný 2011) Vývoj virtuální reality přirozeně zasáhl i geoinformační vědy, kdy 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). Jedná se o: topografická data a analýz 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.

V případě přidání 3. dimenze do kartografie vyvstává terminologický konflikt ohledně toho, co je 3D model a co mapa, popř. co je 3D mapa. Definice se různí napříč literaturou, (Herman 2019) definuje 3D mapu jako: "3D mapu lze definovat jako reprezentaci povrchu Země nebo různých přírodních a socioekonomických prvků ve formě *rea-3D* (využívá binokulárních i monokulárních vodítek) nebo ve formě *pseudo-3D* (využívá pouze monokulárních vodítek – viz. kap. X), s využitím matematického základu. Tento základ zahrnuje geografické nebo projekční souřadnicové systémy a zahrnuje škálu Z (výškové) osy pro vstupní data. Tyto 3D mapy často také zahrnují perspektivní nebo ortogonální grafické projekce.

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 conceptualizacích 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 v rámci virtuálního světa. VGE by tohoto měli dosáhnout skrze kombinaci 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, a VGE tuto technologii integrují. Interakce ve VGE vyžaduje odlišné myšlení než u desktopových počítačů, protože cílem je vytvořit zážitky srovnatelné se skutečným světem. Prostor 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 pracovní prostor 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 případně prostorou 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 jako virtuální atlas (*globe* – ve smyslu celé země) sestavená z masivních dat pozorování Země s více rozlišeními, více časovými obdobími, různými typy dat o Zemi a spřízněnými 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

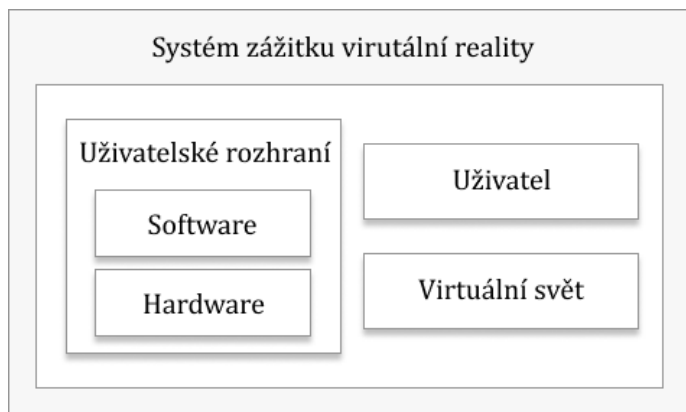
Podobně jako v případě definice virtuální reality, i koncept Metaverse je obtížně definovatelný a jeho význam se mění v závislosti na perspektivě a kontextu. Existuje mnoho různých názorů a přístupů k tomu, co Metaverse skutečně představuje. Jedním z aspektů, který ilustruje nejednoznačnost Metaverse, 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.*

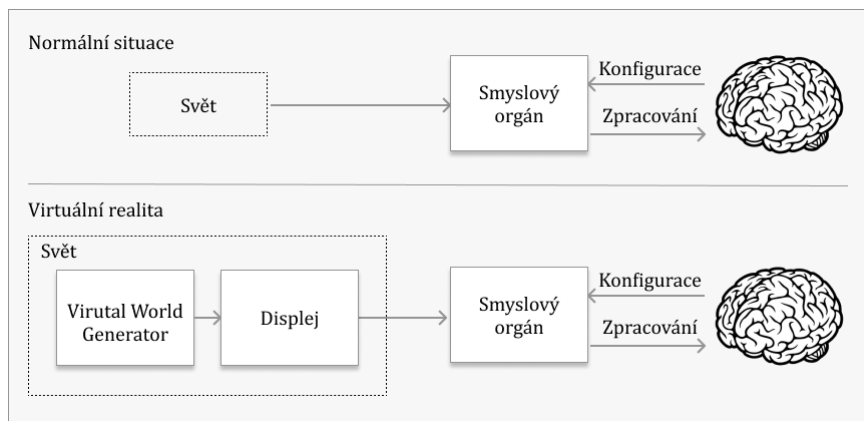
Všechny výše uvedené termíny (VGE, DE, Metaverse) se v různých měřítkách snaží definovat koncept virtuálního světa vytvořeného pomocí současné technologie. Světa, který existuje paralelně se světem „reálným“ a je s ním oboustranně propojen, a to tak že z něj čerpá a zároveň jej zpětně ovlivňuje. Výše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

### 4.3.4.2 Systém virtuální reality

Z obecného hlediska popisuje systém virtuální reality Sherman a Craig a dělí jej na základní komponenty (Software, Hardware, ~~Virtuální~~Virtuální svět, ~~Účastník~~Uživatel) viz. Obr. X. LaValle rozděluje systém virtuální reality na obdobné komponenty (Hardware, Software ~~a~~a ~~Účastník~~Uživatel). Toto rozdělení je patrné i v případě (Coltekin et al. 2020), kde autoři využívají rozdělení 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*, komponenta, která vytváří alternativní prostředí (může být VGE). Následně pak pomocí komponenty displeje, 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)

#### 4.3.14.2.1 Hardware

LaValle dělí hardware na 3 kategorie a to: **displeje** (výstup), **senzory** (vstup) a **výpočetní jednotky** (*processing unit*) (viz. kap. X). Specifický displej pro každý smyslový orgán, na který se daný systém soustředí. Nejdůležitějším je zrak, způsoby, jakými je tento smysl ovládán se liší na základě samotného hardwarového zařízení. V případě HMD se jedná o LED obrazovky

Commented [LH3]: Asi radši uživatel než účastník

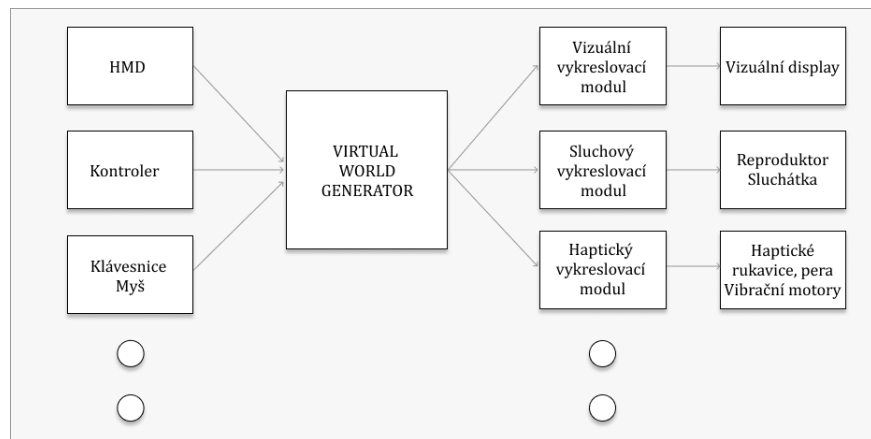
Formatted: Indent: Left: 0", Hanging: 0.53"

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 (*inertial 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 hardware jsou i mechanické vstupy jako klasické klávesnice, myš a jiné ovladače.

Výpočetní jednotka, je klíčová v případě *body-fixed* hardware. Hlavní dělení je tedy v tom, kde je výčetní jednotka, na které je VWG spouštěn, v rámci displeje (HMD) či na externím počítači, popř. jiném zařízení. V případě této práce se nejčastějším případem pak ten, kdy výpočetní jednotka je součástí HMD.

#### 4.3.24.2.2 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. Analýza technologií). Souhrn softwarových komponent potřebných pro vývoj virtuálního zážitku jsou zdůrazněny Obr. X. VWG dostává input z nízko úrovněových 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, kontroléry). (LaValle 2020) Různé typy virtuální reality využívají různé vstupní zařízení (viz. dělení virtuální reality dle míry imerze kap. 4.1) tím pádem i schéma Obr.X nezobrazuje úplný výčet. V případě Imerzní virtuální reality, využívající primárně HMD a kompletně virtuální svět jsou vstupní zařízení z Obr. X považována za hlavní.



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

#### 4.4.0 Účastník - percepce a fyziologie

#TODO - co vybrat? vizuální percepce, imerze, presence??

Nejdůležitějším komponentem v rámci systému zážitku virtuální reality je účastník, proto je nutné rozumět procesu vnímání (percepce) (Sherman, Craig 2019).

**Commented [LH4]:** Může být (překlad by byl fajn - hlavní je to mít ve všech schématech stejné). Viz můj komentář výše k termínům "immersive VR", "desktop VR" - s tímto to souvisí - zejména se sloupcem "input"

**Commented [JH5R4]:** Předěláno do češtiny a stejného stylu - vysvětleno vůči dělení vr podle imerze.

**Formatted:** Normal

**Formatted:** Normal

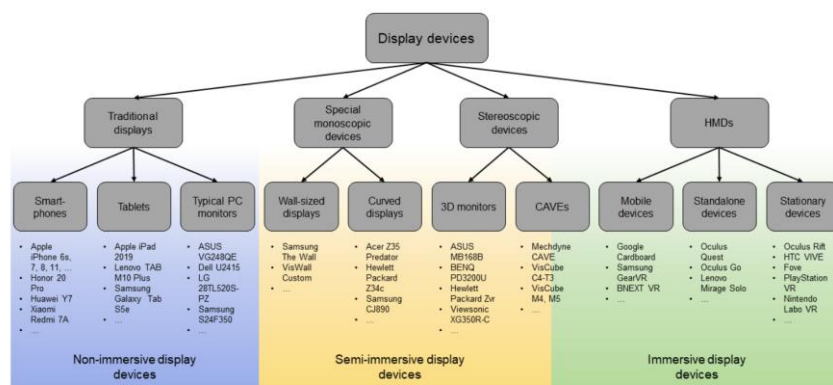
Percepce se uskutečňuje po převodu podnětů smyslovými orgány (a jejich receptory) na nervové impulzy. Výše zmíněné koncepty virtuální reality jsou diskutovány obecně, proto je nutné zmínit, že práce se zabývá především vizuálním vjemem, tedy především koncepty Imerze a Interaktivity budou diskutovány v kontextu vizuálních vjemů.

#### 4.8.4.3 Komponenty systému virtuální reality

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

##### 4.8.4.3.1 Výstup – Output

Zobrazovací zařízení je možné klasifikovat dle míry imerze, kterou poskytují na ne-imerzní, semi-imerzní a plně-imerzní (Coltekin et al. 2020) (viz. Obr. X). 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 zorné 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*, *stationary* a *head mounted*. Nejrozšířenější terminologií pro tento hardware je *head mounted display* termín. Tyto displeje dosahují plného zaplnění zorného pole položením obrazovky velice blízko k očím. Lidské oko však není schopno ostřit na potřebně blízkou vzdálenost (cca 5–10 cm). HMD tedy využívají konvexní čočky položené do jejich fokální vzdálenosti. Díky tomuto se pak obrazovka jeví jako obrovský virtuální obraz, který je nekonečně daleko. (LaValle 2020).



Obr. 5 Dělení HMD, zdroj: (Coltekin et al. 2020) #TODO předělat do češtiny?

V případě mobilních tradičních displejů je možné mluvit o „případné“ imerzi, kterou je možné dosáhnout kombinací s nastavením umožňujícím 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. Výše zmíněné kategorie (Obr.X) jsou tedy v určitých případech propojitelné. Nutné však zmínit, že mobilní zařízení „vylepšené“ o nastavení na hlavu s čočkami se imerzí zdaleka nevyrovná 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 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 implementoval tracking očí (LaValle 2020).

Výstupem není pouze vizuální, ale i haptický, popř. zvukový. Možnosti jednotlivých HMD se i těchto aspektech výrazně odlišují. V Tab. jsou shrnuty klíčové charakteristiky populárních HMD. Pro vývoj aplikace je primárním kritériem jaký způsob snímání popř. interakce (*tracking*) implementuje (viz. kap. Input). V Tabulce nejsou zahrnuty nededikované HMD jako Google Carboard aj., které pomocí čoček umožňují základní imerzi skrze mobilní telefon či jiné zobrazovací zařízení.

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

Název HMD	Typ	Rok výroby	Rozlišení [na oko]	Obnovovací frekvence [Hz]	Tracking	Kontrolery
Apple Vision Pro	Samostatný	Neznámý	3400x3400	90	6DoF - inside-out - oči - obličej - 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, obličej, 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

#### 4.8.24.3.2 Vstup - Input

Důležitým aspektem hardware pro VR je *tracking* (Coltekin et al. 2020). Jedná se o získávání kontinuální informace o poloze a pohybu v rámci reálného světa je klíčovou součástí VR systému. 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

**Commented [LH6]:** To je (zatím stále) limit VR - takže za mně klidně uvést. Doplnil bych to o informaci kolik je cca rozlišení současných brýlí/helem

**Commented [JH7R6]:** Dodělána tabulka

**Commented [LH8]:** Asi by bylo dobré se krátce vyjádřit i prvním dvěma kategoriím v tomto schématu... (ne-imerzivní a semi-imerzivní). Stačí ale fakt krátce = po jednom odstavci

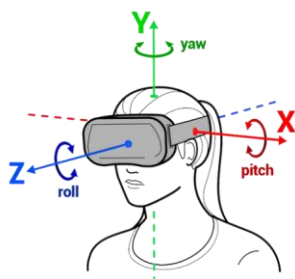
**Commented [JH9R8]:** Pokusil jsem se o nějaké vyjádření k tomu, že vlastně mobil s google cardbordem je HMD samo o sobě.

2. Ostatní části těla – obličej, 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.X**. Obecně 3 DoF znamenají rotace na daném místě a 6 DoF znamená rotace na daném místě a pohyb v prostoru (translate). 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 kamerami v rámci HMD, kdežto *outside-in* vyžaduje externí kamery, které snímají zařízení a tím určí 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 (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.



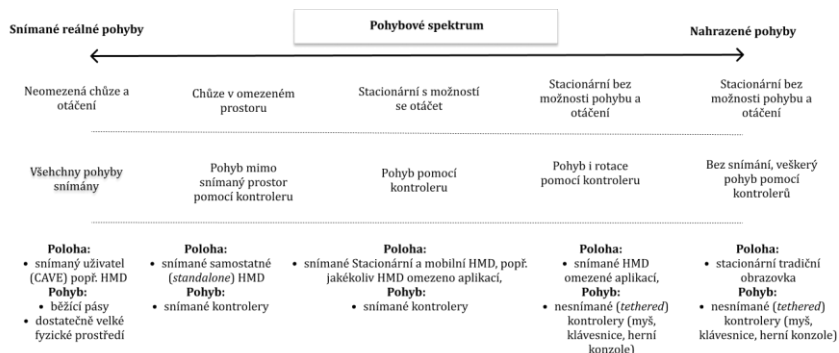
**Obr. 6 Stupně volnosti – Degrees of Freedom – DoF (zdroj: (Dupin 2016))**

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 důvodů

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



Obr. 7 Pohybové 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 světě 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.
- **Teleportaci** – 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 **selekcce**, **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. X). Dle tohoto API lze vstupy rozdělit na dvě hlavní kategorie, jimiž 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 vyslán 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 vyslán z ovladače a interakce spouštěna pomocí tlačítek (*pointer based*). V případě neexistence ovladačů je vhodně 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 tlačítek na klávesnici). Řešením je převod klávesnice, ovladačů aj. do virtuálního prostředí nebo i vstup pomocí snímání pohybů rukou popř. ovládaní pomocí hlasu (Vicente et al. 2022).



Obr. 8 Vybrané příklady řešení interakce ve VP. zdroj: (Mozzila Corporation 2023a; Ravasz 2019)

#### 4.4 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 počítčích, které jsou inicializované podněty ze smyslových orgánů. Více počítků pak tvoří vjem.

(Chloupková 2007) Pro VR je klíčová percepce (vnímání) vzdálenosti a měřítka, pohybu, barvy a následně jejich kombinace (LaValle 2020).

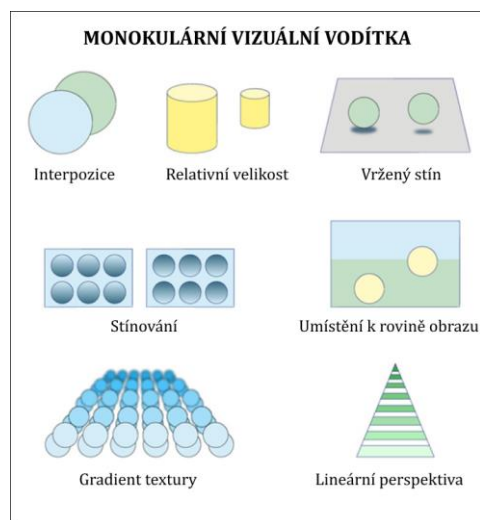
#### 4.8.34.4.1 Percepce vzdálenosti

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

Dynamická vodítka jsou pak **paralaxa pohybu** – kdy pohyb vzdálenějších objektů po sítnici oka je pomalejší (Chloupková 2007), **mizení (deletion)** a **přirůstání (accretion)** – při

nepřímých pohybech se objekty ve různých vzdálenostech jeví, že se relativně pohybují jeden k druhému. Tento jev se nazývá "mizení", když objekt vzadu postupně mizí za objektem vpředu. Označuje se jako "přirůstání", když objekt vzadu vystupuje zpoza objektu vpředu (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, **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)

Znalost těchto procesů je klíčová pro tvorbu VR prostředí, jelikož může snadno dojít k neshodám v reálné velikosti či vzdálenosti objektů, tedy špatnému vnímání měřítka, kvůli špatné interpretaci prostorových vodítek. Příkladem neshody může být (*vergence accommodation mismatch*). Dalšími příčinami neshod je pak nedokonalý tracking hlavy uživatele, kdy výrazná latence působí opoždění zobrazení. Problematické je i když tracking sleduje pouze orientaci hlavy a tím pádem znemožňuje použití paralaxy pohybu. (LaValle 2020) zmiňuje fakt, že monokulárních vodítek by mělo při tvorbě ~~virtuálního~~ virtuálního prostředí být využíváno co nejvíce.

#### 4.8.4.4.2 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 vizuálního a rovnovážného aparátu nazvaný *vection*. 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).

*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 (VOR), 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 dlouhý a uživateli se pak jeví jako přeskakující („judder“) namísto plynule se pohybujícího. Jelikož vysoké FPS hodnoty jsou 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)

#### 4.8.5.4.3 Význam pro tvorbu VR prostředí

V případě, že nejsou všechny smyslové vjemy nahrazeny virtuálními vstupy nebo pokud vstupy nejsou dokonalé (nejsou v konfliktu s lidskou fyziologií) dochází ke konfliktům vnímání. Nejvíce problematickým se dlouhodobě jeví vekce a to 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

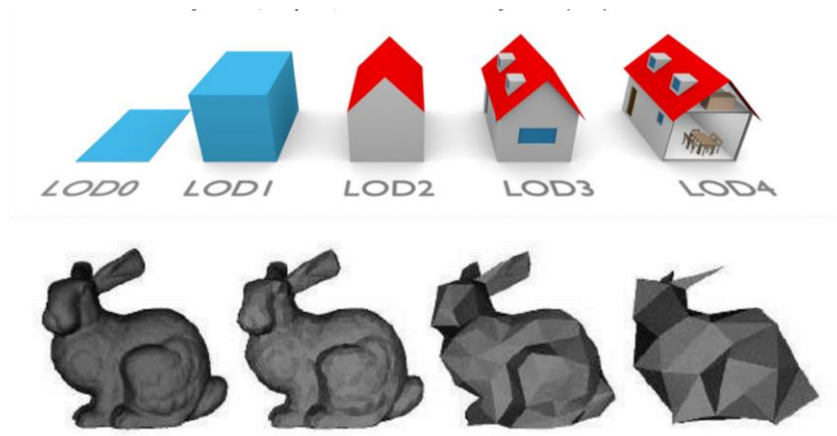
zkušenostmi. Tyto konflikty pak vedou ke špatným či nechtěným interpretacím, popř. k nevolnosti a únavě. (LaValle 2020)

## 4.94.5 3D modelování koncepty a principy

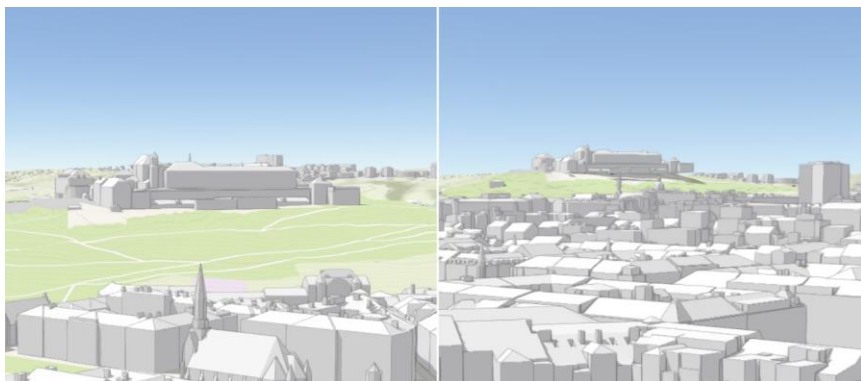
### 4.9.14.5.1 Level of Detail, Level of Realism, Level of Abstraction

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 (Obr. X). S rostoucí vzdáleností pozorovatele od objektu se používají verze s nižším stupněm detailu, což snižuje výpočetní nároky a zlepšuje rychlost 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. X) z něž je patrné, že aplikovaný LOD přístup na data DMT vede k tomu že budovy (hrad Petrov) jsou při pohledu z dostatečné vzdálenosti nevhodně zobrazeny nad zjednodušeným terénem. Případně řešení tohoto problému pak poskytuje práce (Semmo, Döllner 2014), kde autoři navrhnou interaktivní změnu LOA dle akcí uživatele.

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)

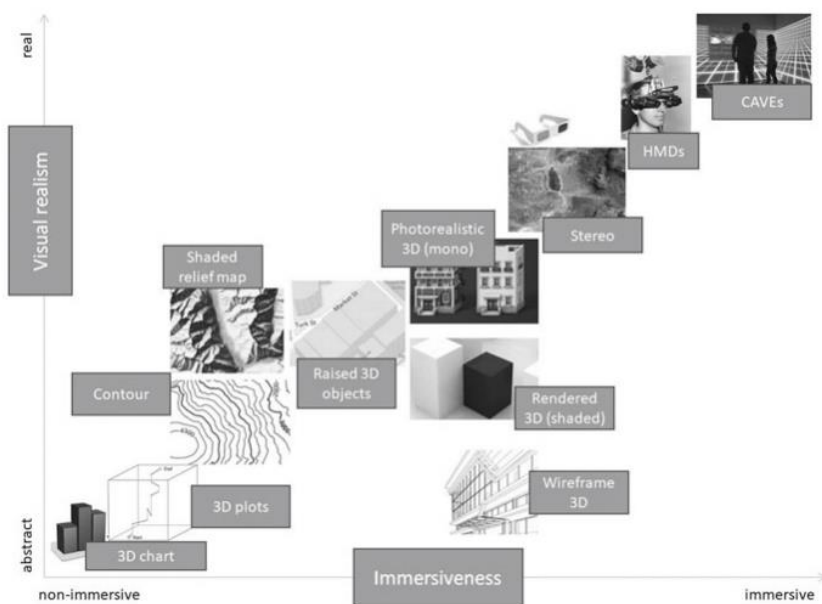


Obr. 10 Příklad konceptu sémantického LOD v případě specifikace CityGML – nahoře; v případě redukce vertexů – dole. (převzato z: (Biljecki, Ledoux, Stoter 2016) a (Ghulam et al. 2013))



**Obr. 11** Snímky obrazovky z aplikace 3D model města Brna. Hrad Petrov pohledu ze směru Vila Tugendhat – Petrov. vlevo – terén je vykreslován v plném rozlišení; vpravo – terén je vykreslován zjednodušeně, model hradu je posazen nad terénem. (KAM Brno 2023)

Jako důležitou problematiku zmiňuje (Coltekin et al. 2020) 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 (Coltekin et al. 2020). Při zanedbání některých z těchto aspektů v návrhu vizualizace se pak jedná o snížení míry imerze.



**Obr. 12** Klasifikace 3D vizualizací, na základě LOR a míry imerze. Zdroj: (Çöltekin, Lokka, Zahner 2016)

Další z problematik je pak tvorba virtuálního obsahu. Vytvoření obsahu s vysokým LOD a LOR zahrnuje řadu komplexních operací (Coltekin et al. 2020). Za hlavní úskalí se považuje kombinace množství vstupních dat (LiDAR, tomografie, stereofotogrametrie aj.), manuální práce při samotném modelování jako např.: segmentace textur od povrchů, tvorba topologicky správných 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.

#### 4.9.24.5.2 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 rychlost 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, **možnosti uchovat informaci o geolokalizaci**, 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). Jelikož je možných formátů pro 3D geografickou vizualizaci mnoho budou pro analýzu budou vybrány pouze relevantní pro tuto práci. Z nich pak budou vybráni kandidáti pro tvorbu pilotní aplikace, kteří budou podrobněji popsáni. **#TODO – zmínit geoformáty, ale nejt moc podrobně, hlavně gltf**

V **Tab.X** Sloupec „web“ udává, jak často daný datový formát nalézá uplatnění na internetu v současném kontextu. Sloupec indikuje, zda webové využití formátu je omezené (např. licencovaným softwarem), popř. méně běžné vzhledem k modernějším technologiím a standardům. Toto dělení je utvořeno na základě autorovy znalosti není tedy výsledkem exaktního výzkumu, tudíž by nemělo být bráno jako stoprocentně směrodatné. 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).

Formát **gltf** přezdívaný *JPG pro 3D*, jedná se o otevřený formát vytvořený skupinou Khronos. Jedná se o formát určený pro sdílení 3D scén. Může být ve dvou formách – 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.

V kontextu geoprostorových dat, gltf obsahuje 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. Formát je podporován v mnoha desktopových i webových řešeních (Khronos Group 2017). Formát GLTF je schopen obsáhnout kompletní 3D scénu se všemi komponenty (**viz. kap. X**).

Pro velké objemy primárně geografických dat v 3D scénách vytvořila firma 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 **dětských nódů**. 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)

**Tab. 2 Seznam relevantní datových formátů umožňující 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

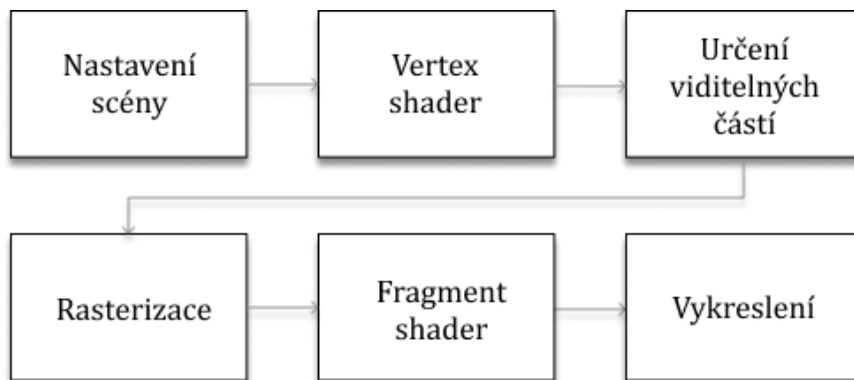
## 4.10.4.6 3D grafika

### 4.10.14.6.1 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 . . .) 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 renderována. Dále následuje nastavení osvětlení, objektů, materiálů aj. komponentů **3D scény (viz. xx)**.
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. částí 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 "fragments". Při rasterizaci se určuje, které body nebo vrcholy 3D objektu spadají do tohoto pixelového rastru na obrazovce. Atributy vertexů v rámci jednoho pixelu jsou interpolovány. Interpolace probíhá zpravidla na attributech 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. 13 *Graphics rendering pipeline*. vlastní tvorba dle (Ghayour, Cantor 2018; Ariën 2017; Sherif 2018).

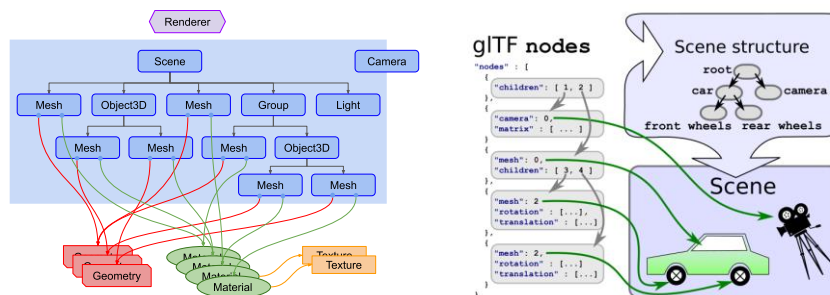


#### 4.10.24.6.2 Komponenty 3D vizualizace

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

**Scéna:** Scéna je jedním z klíčových prvků virtuální reality a 3D vizualizací a představuje digitální prostředí, ve kterém se odehrává uživatelský zážitek. Scéna funguje jako základní stavební blok, do kterého umístíme všechny objekty, které chceme zobrazit a s nimiž chceme interagovat. Struktura scény je implementována pomocí hierarchické stromové datové struktury obdobně jako je DOM v prohlížečích. Tato struktura se nazývá graf scény (*scene graph*). Tato struktura se využívá napříč 3D softwarem např. formát gltf, Blender nebo three.js. Scéna zpravidla obsahuje druhy objektů, které jsou Světla, Objekty, Meshe. **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)

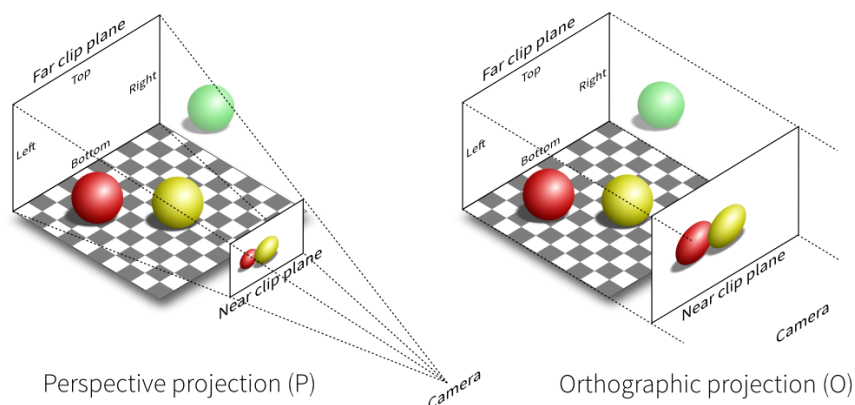


Obr. 14 Příklad obecného grafu scény knihovny three.js. a formátu gltf. (three.js Contributors 2023c; Khronos Group 2023a)

**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. V rámci virtuální kamery jsou pak klíčové parametry (Dunn, Parberry 2011; three.js Contributors 2023a):

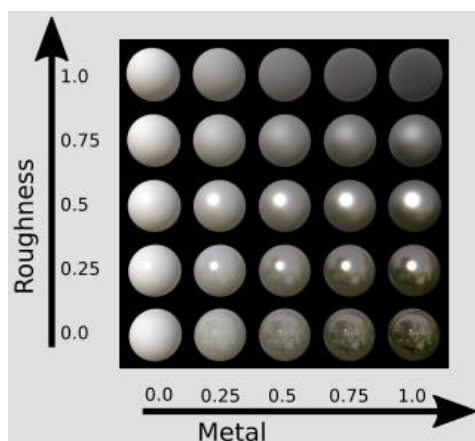
- **Zorný úhel:** Zorný úhel (*FOV - field of view*) určuje, jak široký úhel zobrazení kamera má.
- **Poměr stran** (*aspect ratio*): Poměr stran definuje proporci mezi šířkou a výškou obrazovky nebo zobrazovacího okna kamery. Správný poměr stran je důležitý pro zachování proporcí objektů ve scéně.

- **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 nebudou zobrazeny (viz. *frustum culling*).



Obr. 15 Příklady perspektivní a ortografické kamery. (three.js Contributors 2023a)

**Materiály:** „Parametrizovaný přibližný 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 světlem 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 povrchy, Phongův – jakým způsobem se světlo odráží od lesklých povrchů 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 (Odráživost). Zároveň PBR model je schopen 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 světla. Tedy při simulaci žárovky o 100 wattch není možné osvětlovat prostor větší než standardní místnost (Discover three.js Contributors 2023).



Obr. 16 PBR Metallic a Roughness parametry. Zdroj: (Chow 2018)

**Texture:** Texture 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 texture využity k definici vlastností, jako jsou barva, lesk, hrubost a další optické charakteristiky. Textura je obyčejná 2D bitmapa, kterou je nutně 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 textura modifikuje normálový vektor povrchu na úrovni jednotlivých pixelů. Tímto tedy 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).

**Osvětlení:** Osvětlení ovlivňuje, jak se objekty ve scéně zobrazují. To může zahrnovat sluneční světlo, umělé světlo, stínování a další efekty. Obecně se světla v rederovacích enginech 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 slábne se vzdáleností od něj.
- **Spot Light** – Obdobné jako Point Light, světlo je ve tvaru kužele, 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 v je možné dosáhnout pomocí techniky stínovacího mapování (shadow maps), kdy pro každý zdroj světla, který vytváří stíny jsou všechny objekty schopné

vytvářet stíny vykreslovány z pohledu daného světla. Tento způsob vede k opakování vykreslení celé scény (*draw calls*) 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*).

#### 4.10.34.6.3 Výkon

Při vývoji VR aplikace pro web. Je nutné mít na paměti parametry ovlivňující virtuální zážitek. 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 využití 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 statisců.
- **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. V rendering engine three.js textura zabere *šířka \* výška \* 4 \* 1.33* bajtů operační paměti. Což pro texturu o rozlišení 1024x1024 znamená 5.19 MB operační paměti. (three.js Contributors 2023e)
- **Počet světel** – Větší počet dynamických světel znamená větší počet vykreslení a větší výpočetní náročnost pro simulaci osvětlení.

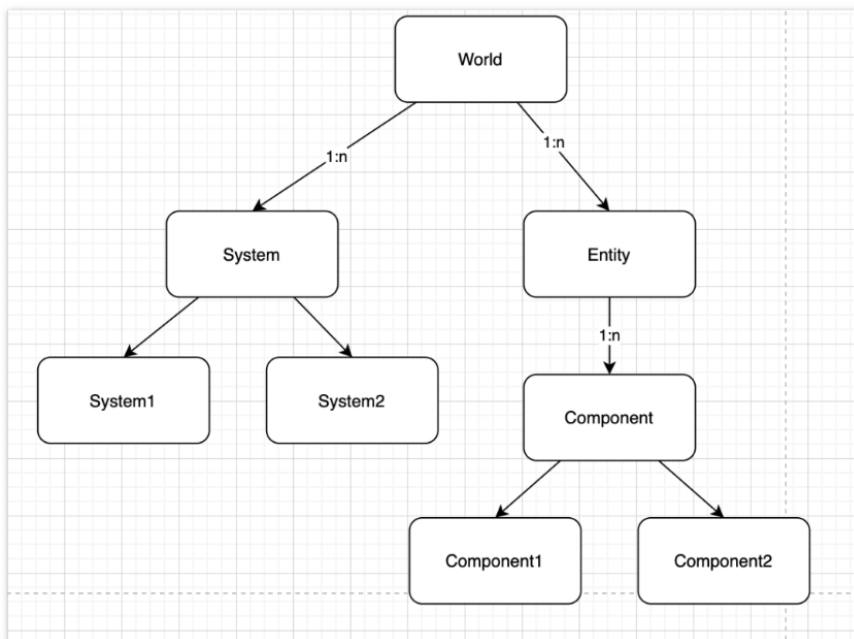
#todo Techniky optimalizace – stručně – až je použiju 😊.

Průhlednost materiálů je náročná na počet vykreslení + v renderovacích enginech často nastává problém určení, které objekty zakrývají jiné.

#### 4.10.44.6.4 Vývoj

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 světě 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. Základními pravidly ECS je, 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 entitou může být např. miniatura 3D modelu města, které je přiřazeno komponenty, které umožňují rotaci pomocí kurzoru aj.

Tento přístup využívá řada herních engineů (Unreal Engine, Unity, Godot) a také webové řešení jako Babylon.js a A-Frame popř. řešení jako Mozilla Hubs, Third Room a Ethereal Engine ([podrobný rozbor technologií viz. X](#)). (Stapley 2022; Ford 2017; Mozilla Hubs 2022)



Obr. 17 Architektura ECS návrhového vzoru. Zdroj: (Stapley 2022)

## 5 ANALÝZA TECHNOLOGIÍ

Stěžejní kapitola práce se zaměřuje převážně na analýzu technologií, které mohou tvořit vhodný tzv. *Virtual World Generator* (viz. kap. **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 technologii 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. Tudíž i za účelem tvorby aplikace pro vizualizaci dat na webu ve virtuální realitě je potřeba souhra více technologií, které umožní pořízení dat, úpravu dat, vizualizaci, interakci a následně publikaci. Nejedná se tedy o jednotlivou technologii, ale souhru více tzv. *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 je možné zjednodušit dle úkolů na:

### 1. Získání dat pro zobrazení – SLAM

- a. Lidar, fotogrammetrie, tradiční mapování (textury)

### 2. Zpracování dat

- a. Vytvoření 3D objektů (mesh), zpracování textur, vytvoření 3D scény

### 3. Vizualizace dat

- a. Zobrazení dat pomocí vybraného vykreslovacího enginu, aplikační logika – implementace uživatelské implementace s daty

### 4. Publikace dat

- a. Zvěřejnění vizualizace v rámci internetu, testování, optimalizace

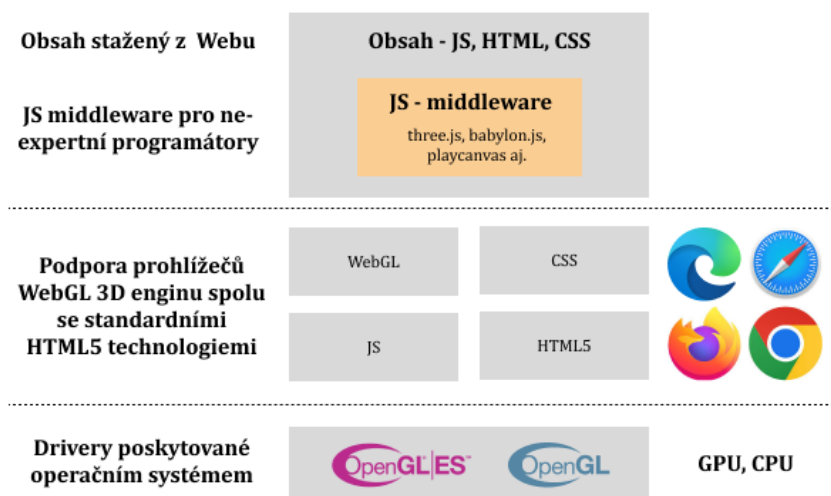
### 5.1 Taxonomie technologií

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).

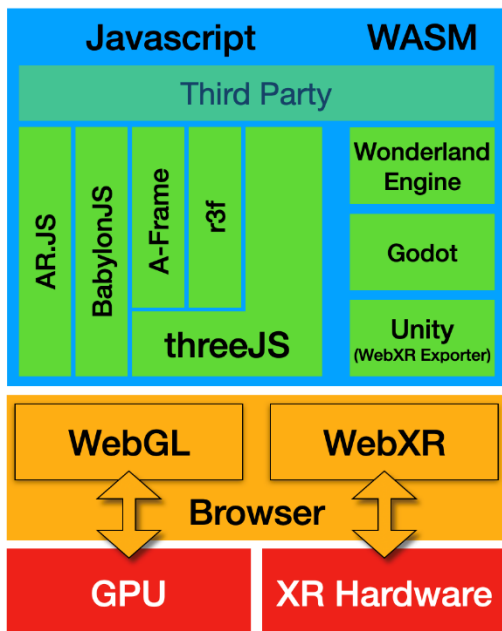
Pro dosažení daného případu užití je v mnoha případech využíváno technologií napříč všemi zmíněnými kategoriemi. Zde je nutné zmínit, že pro vývoj výsledné webové vizualizace neznamena využití pouze webových technologií. Tento fakt je prominentní především při přípravě a zpracování geoprostorových dat. 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 engine a export WebGL aplikace skrze WASM (*Web Assembly*).
- Vývoj skrze dedikovaný WebXR engine.

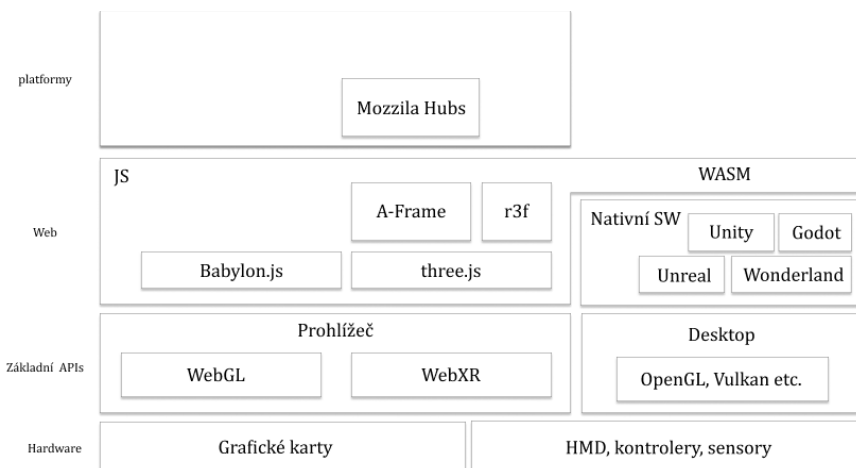
Pro účely geoprostorové vizualizace je možné vyloučit postup č. 1, jelikož se jedná o příliš nízko-úrovňový vývoj. V rámci této práce však bude role základních API teoreticky představena (viz. kap. X). Hlavním zaměřením je pak postup č. 2, kdy vývoj probíhá na základě standardních webových technologií. 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 (viz. Needle engine). Postup č. 4 je uživatelsky nejpřívětivější cestou, jelikož umožňuje tvorbu VP v rámci GUI a přímou kompatibilitu s WebXR API, jedná se však o nový produkt stále ve vývoji, jelikož samotná WebXR API je stále novou specifikací.



Obr. 18 Taxonomie technologií umožňujících vykreslení 3D grafiky na webu. Upraveno dle: (Khronos Group 2018)



Obr. 19 Taxonomie [webových](#) [webových](#) technologií umožňujících tvorbu [virtuálních](#) [virtuálních](#) prostředí. zdroj: (Godber 2022) - [#todo dodělat o vlastní - předělat na vlastní relevantní](#)



Obr. 20 Taxonomie [webových](#) [webových](#) technologií umožňujících tvorbu [virtuálních](#) [virtuálních](#) prostředí. zdroj: (Godber 2022) - [#work in progress](#)

Obr. X a Obr. X podávají obecný přehled o ekosystému technologií, kterých je třeba pro vykreslování 3D grafiky a tvorby virtuálních prostředí na webu.



## 5.2 Webový vývoj

Volba webového prostředí přináší jisté benefity, ale i překážky při tvorbě VR aplikací. Primárním benefitem webové vizualizace je dostupnost (*availability*) a přístupnost (*accessability*) oproti desktopovým aplikacím. 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 technologií. 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 2-D vektorovou grafiku na webu
4. JavaScript – vysokoúrovňový jazyk umožňující interakci s obsahem a vzhledem

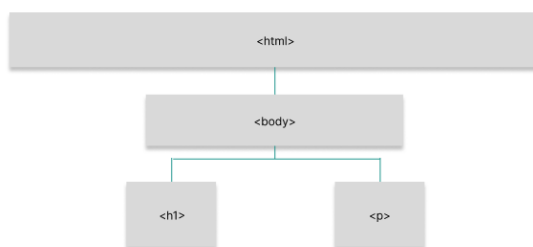
Mezi webové standardy je pak volněji možné zařadit i klíčové API (DOM, Fetch API, WebGL, WebXR aj.), které umožňují snazší vývoj pro webové prostředí.

### 5.2.1 Web APIs

#### 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 daný 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. 21 Zdrojový kód HTML a DOM struktura. upraveno podle: (Řeháček 2020; W3Schools 2023)

#### WebGL API

Jedná se o nízko úrovněvé Javascriptové aplikační rozhraní, které umožňuje vysoko výkonnostní vykreslování interaktivní 3D a 2D grafiky v rámci kompatibilního webového prohlížeče. Díky tomu, že WebGL následuje implementaci OpenGL ES (*OpenGL for Embedded*

Commented [LH10]: Myslím, že většinou se to píše dohromady

Commented [JH11R10]: .

Systems<sup>1)</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 2023b). Pro vykreslování využívá WebGL jazyk GLSL, který popisuje způsob, jakým je daný objekt vyobrazen (*shader*). Základní funkčnost WebGL dále rozšiřují JavaScriptové knihovny (viz. kapitola X), 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.

### 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. Jedná se o hardware agnostické imperativní API, které umožňuje jednotný přístup k vytváření virtuálních zážitků pro mobilní i desktopový VR hardware. WebXR API je založeno na OpenXR<sup>2)</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 (3d render) 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):

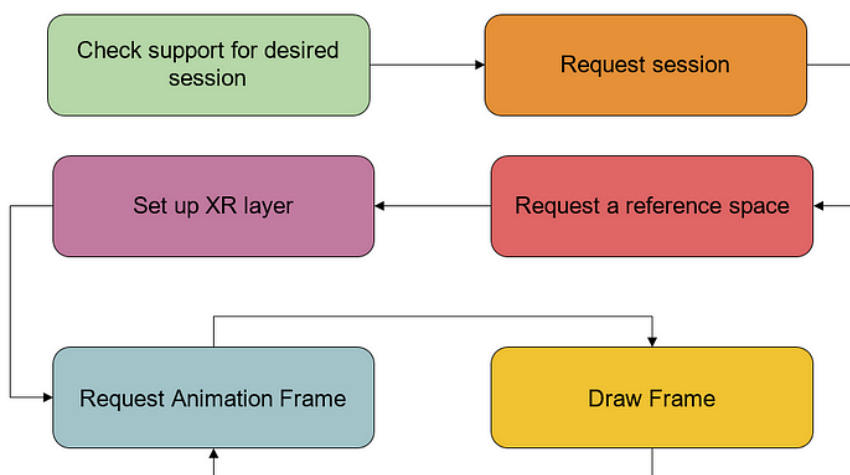
1. Dotaz na to, zdali je XR podporováno (prohlížečem / zařízením) a jaký typ relace je podporován, popř. vyžadován.
2. Pokud je podpora k dispozici, informovat uživatele o dostupnosti vyžadované XR funkcionality např. v podobě přidání tlačítka umožňujícího spuštění virtuálního prostředí.
3. Uživatelská událost (*event*) signalizující, že uživatel má zájem o spuštění XR relace.

<sup>1</sup> WebGL je založen na specifikaci OpenGL ES 2.0 s cílem maximalizovat přenositelnost na mobilní zařízení namísto OpenGL, který je pro desktopová řešení.

<sup>2</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.

- Požadavek na relaci, kde je možné specifikovat typ (*inline*, *immerive-vr*, *immersive-ar*) a vlastnosti (např. možnosti pohybu uživatele poskytnuté daným zařízením viz. kap. **Input** *unbounded*, *local* aj.), které by požadovaná relace měla splňovat.
- Vytvoření vykreslovacího cyklu, který obnovuje senzorická data (poloha, pohyb atd.) a který generuje snímky na obrazovku zařízení.
- Vykreslování, dokud uživatel nespustí událost ukončení XR relace.
- 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) viz. kap. X skrze XRWebGLLayer.



Hlavním přínosem WebXR je standardizace práce se vstupními zařízeními. WebXR definuje *XRInputSource* objekt, který poskytuje informace o vstupním zařízení. Základní schopnosti vstupních zařízení jsou:

### WebGPU API

WebGPU je dalším krokem v evoluci webových grafických API, tedy zamýšleným následníkem WebGL. Jedná se o experimentální aplikační rozhraní, které umožňuje vysoko výkonnostní 3D a 2D vykreslování na webu. WebGPU je nízkourovňové API, které poskytuje programátorům přímý přístup k hardwarové akceleraci na grafických kartách. Rozdíl 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).

Formatted: Font: Not Bold, Not Highlight

### 5.2.2 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. X**. V rámci tabulky jsou zahrnuty prohlížeče, které implementují WebXR API. Nutné zmínit, že tabulka neobsahuje prohlížeče specializované na AR (*Magic Leap Helio*). Pro podporu WebGL není nutná podrobná analýza, jelikož všechny zmíněné prohlížeče v **Tab. X** plně podporují WebGL 1.0 a WebGL 2.0 (Can I Use 2023a). Podpora WebXR není tak rozšířená, tudíž přehled je zobrazen v **Tab. X**.

**Tab. 3 Podpora WebXR Device API ve verzích vybraných prohlížečů.** – <sup>1</sup>WebXR Device API je velice 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	-	-

Formatted: Don't keep with next

### 5.2.3 Vývojářské nástroje

#todo – představit emulátory HMD vstupů – WebXR rošíření pro Firefox, Oculus emulator pro Chrome. Propojení s HMD skrze USB.

## 5.3 Specifikace požadavků pro technologie

# TODO – jak tohle vůbec napsat, kam to dát

Obecnou metodikou využívanou v softwarovém inženýrství je specifikace uživatelských požadavků na výslednou aplikaci. Hodnocení technologie pak tedy závisí na tom, zda umožňuje dosažení definovaných požadavků. Metodika specifikace požadavků byla převzata z práce (Herman 2014) a modifikována pro účel hodnocení aplikací umožňující tvorbu virtuální reality.

### 5.3.1 Funkční požadavky

Jedná se o formulace služeb / funkcí, které by měl systém poskytovat, specifikaci reakcí systému na dané vstupy a chování systému v daných situacích. V některých případech se funkční požadavky rovněž zaměřují na explicitní definici činností, jež by systém neměl vykonávat. (Sommerville 2016)

#### Orientace ve scéně

Pro úspěšnou tvorbu VR aplikace je nutné, aby technologie podporovala především způsob pohybu a orientace na základě trackingu polohy a orientace zobrazovacího, popř. vstupního zařízení.

#### Pohyb scénou

Technologie by měla podporovat různé možnosti pohybu 3D scénou. Měla by podporovat poziční *tracking*, tedy proporčně převádět relativní polohu zobrazovacího zařízení od počáteční polohy v rámci virtuálního prostředí. Ve VR prostředí je dále běžným požadavkem možnost zrychleného pohybu nejčastěji pomocí teleportace skrze ukazovací paprsek. Technologie by měla podporovat způsoby pohybu pro různé typy vstupních zařízení (klávesnice, myš, kontrolery, aj.)

#### Zobrazení

Technologie by měla umožňovat zobrazení na úrovni imerze jak Desktop VR tak Immersive VR. Tedy pokud uživatel přistoupí k výsledné aplikaci pouze skrze tradiční display, měl by být schopný aplikaci používat stejně jako, když přistoupí pomocí HMD.

#### Geoprostorová data

Technologie by v ideálním případě měla nativně podporovat geoprostorová data. Měla by mít možnost prostorové geolokalizace.

### 5.3.2 Mimo-funkční požadavky

Jedná se o omezení služeb nebo funkcí, které systém nabízí. Zahrnují časová omezení, omezení vývojového procesu a omezení stanovená normami. Nefunkční požadavky se často vztahují na celý systém, spíše než na jednotlivé funkce nebo služby systému. (Sommerville 2016)

#### Cena

Cenu vývoje je možné definovat pomocí nákladů časových, finančních. Pomocí nich lze následně hodnotit jednotlivé technologie, a to skrze finančních nákladů na použitý software (desktopová řešení) a data. (Herman 2014) Ačkoliv je v poslední době i v ČR sentiment zpřístupňování dat bez poplatků, 3D data jsou v mnohých případech stále proprietární záležitostí. (ČÚZK 2023) Časové náklady je primárně pracnost vývoje aplikace pomocí dané technologie. (Herman 2014)

#### Dokumentace

Technologie by měla mít extenzivní a srozumitelnou dokumentaci za účelem snadného vývoje.

#### Výkonnost

Technologie by měla dosahovat dostatečného výkonu, tak aby byla schopná vykreslovat dostatečné množství snímků, aby nedocházelo k snížení imerze. Tento požadavek je poměrně těžké hodnotit u samostatných technologií, jelikož závisí nejen na softwarovém řešení, ale často více na hardwarovém zařízení, popř. na platformě (prohlížeči). Jedním z konkrétních východisek pro výběr technologie je skutečnost, že technologie musí být založena na WebGL.

#todo - jak budu měřit, standardní scéna s povrchem budovou a nějakým objemem tematickým?

#### **Přístupnost**

Technologie by měla umožňovat přístup z co možná nejvíce zobrazovacích (výstupních) zařízení. V tomto případě se jedná o kombinaci podpory technologie v rámci webových prohlížečů a zároveň podpory těchto prohlížečů na vybraných hardwarových zařízeních.

#### **Kompatibilita**

Kompatibilita je v tomto případě úzce spjata s přístupností. Jedná se o množství podporovaných (kompatibilních) vstupních zařízení. Technologie by teda měla podporovat běžná vstupní zařízení. #todo - specifikovat

#### **Interoperabilita**

Technologie by měla podporovat interoperabilitu mezi dalšími systémy, měla by být tedy dostatečně modulární pro její použití spolu s dalšími technologiemi. Interoperabilita v případě geoprostorového kontextu je také na úrovni dat, tedy technologie by měla podporovat standardizované formáty dat a přístupů k jejich zpracování, popř. konverzi.

#### **Imerze**

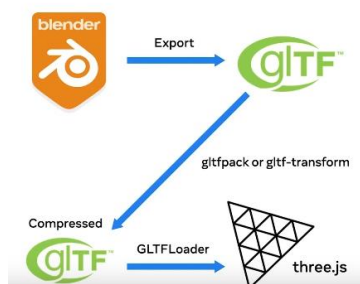
Technologie by měla podporovat Imerzní VR, tedy možnost vizualizace skrze HMD zařízení. Technologie by měla podporovat WebXR.

### **5.3.3 Existující řešení**

**Project Flower Bed** – three.js, custom ecs lib, custom mesh instancing and collision geometry generation and raycasting. 3D content: Geometry – GLTF, Texture - KTX2 Textures format.

Blender – scripting – level editor – custom properties to individual meshes and attach json data as attributes directly to the nodes in GLTF format – read those properties in engine to attach gameplay behaviours ?? HOW

Creating colliders by scripting - ...



Komprese - gltf-transform, gltfpack

### **5.4 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 klíčová pro existenci virtuální reality na webu.

#### 5.4.1 Herní enginy

Tradičně se jedná o desktopové aplikace specializované pro vývoj počítačových her, popř. interaktivních prostředí. Primárním zaměřením je vývoj her a specializovaný export binárních spustitelných souborů specializovaných pro danou platformu. 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 pomocí do WebAssembly<sup>3</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. 4 Herní enginy s podporou pro web export.**

Jméno	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	GScript (podobný Pythonu), C#, C++	WASM	Ano	DAE, glTF, OBJ, JSON a další	bin, DAE, glTF a další	MIT Licence

Jelikož desktopové herní enginy jsou založené na exportu skrze WASM není možné rychlé prototypování při vývoji. Rychlost iterací pro webový vývoj v Unity se snaží řešit software *Needle Tools*, (viz. kap), který poskytuje propojení práce v Unity Editoru se virtuálním prostředím na webu, skrze transformace Unity scén do renderovacího engineu three.js s využitím jazyka TypeScript pro tvorbu komponent.

Tyto herní enginy jsou optimalizovány pro tvorbu tradiční Desktop 3D imerzních prostředí a zážitků. Wonderland Engine je herní engine optimalizován pro tvorbu virtuálních zážitků. Jedná se o desktopový editor a webový javascriptový runtime. Wonderland engine je vyvinut především pro tvorbu 3D na webu spolu s optimalizací výkonu, která je pro web důležitá.

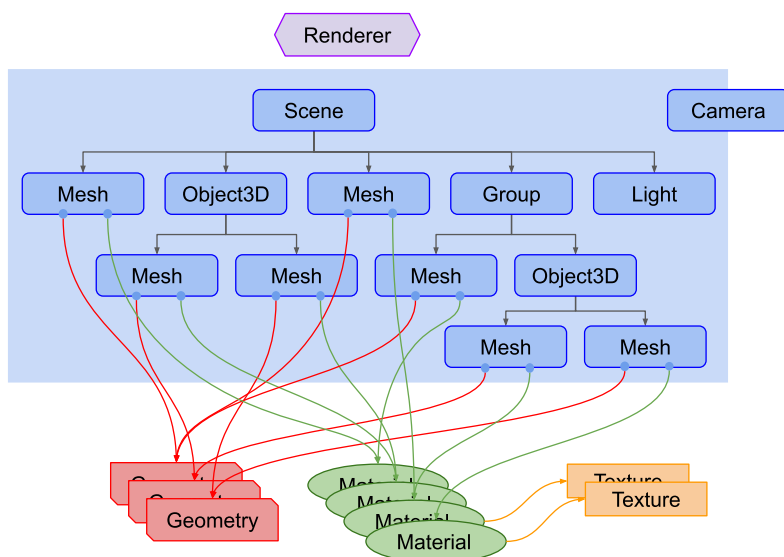
#### 5.4.2 Rendering enginy

Vysokoúrovňové knihovny jako Three.js, Babylon.js, PlayCanvas, SceneGL, PhiloGL a další umožňují vytvářet složité 3D animace, které jsou připraveny k zobrazení v prohlížeči, bez velkého úsilí, které by bylo vyžadováno při vytváření samostatné (nativní WebGL) nebo tradiční aplikace s použitím pluginů.

<sup>3</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.

### Three.js

Jedná se o Javascript knihovnu, tvořící abstrakci pro práci s WebGL. Knihovna byla vytvořena Ricardem Cabellem (Mr. Doob) a je publikována pod MIT Licencí. Jedná se o velice populární knihovnu (více nežli 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 vytváří vlastní ekosystém frameworků a implementací, které rozšiřují její funkcionalitu např. r3f – tvorba 3D UI, aframe – systém pro tvorbu VP a následně i konkrétní implementace v případě geoprostorových informací např. qgis2three.js plugin, ITowns popř. 3dbag-viewer aj.



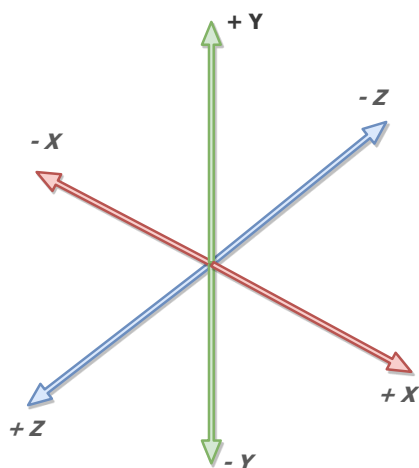
Hlavním komponentou Three.js je Renderer, který při poskytnutí Scény a Kamery umožní skrze WebGL vykreslit část 3D prostředí, které je v záběru kamery jakožto 2D obraz v rámci <canvas> HTML elementu. Hlavní strukturou Three.js je graf scény, který obsahuje objekty (viz. Obr.X). Stromová struktura určuje polohu a orientaci objektů, poloha objektu se určuje relativně k jeho rodičovskému objektu. **Kamera** může být mimo scénu, jelikož může mít danou, popř. interaktivní polohu, zároveň pokud má kamera rodičovský objekt a ten mění svou polohu v prostoru, kamera jej bude následovat. Specifika dalších komponent, jsou obdobná jejich obecnému popisu v (kap. X Komponenty 3D scény).

Hlavní zaměření Three.js je vykreslování grafiky a je vyvíjena s úmyslem jejího propojení s dalšími moduly / knihovnami pro dosažení účelů specifické aplikace. V případě VR existují frameworky jako Aframe, který přidává k three.js deklarativní strukturu práce s komponenty skrze HTML. Dalším frameworkem je r3f (*React three fiber*), jedná se o wrapper kolem three.js,



který umožňuje interakci s three.js skrze komponenty knihovny React<sup>4</sup>. (Meta Developers 2022).

Geolokace v three.js není inherentně implementována. Využívá 3d kartézské soustavy kdy střed scény je (0,0,0) a jednotkou je metr. Jedná se o standard pro vykreslovací enginy a jiné 3D softwary. Orientace os se však může lišit. Každý z objektů, který je v rámci scény má svůj lokální kartézský souřadnicový systém. TRS (translate, rotate, scaling) je definována v souřadnicovém systému otcovského nódu v rámci grafu scény.



Obr. 22 Orientace os souřadnicového systému three.js (Discover three.js Contributors 2023)

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*, pomocí níž je vykreslováno 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. X). #todo – add obrázek z game development book.

Three, PlayCanvas, Babylon.js,

### Babylon.js

Typescript based engine similar to three.js.

### PlayCanvas

<sup>4</sup> React, nebo také React.js, je JavaScriptová knihovna pro tvorbu uživatelských rozhraní (UI) ve webových aplikacích. Jedná se o komponentový framework, který umožňuje vytvářet znovupoužitelné a deklarativní komponenty.

editor based rendering engine with similar workflow as desktop game engines

#### **Wonderland engine**

Editor based engine with inbuilt compatibility with WebXR

### **5.4.3 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 neimmerzní 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 neimmerzní VR. Na základě takto publikovaných scén je následně možné vyvíjet webové aplikace s přidanou funkcionalitou pomocí **ArcGIS JS API** (ESRI 2023a; KAM Brno 2023). Immersive VR úroveň 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 funkcionalita 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 funkcionalitu. 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 **qgis3dthree.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 WebXR API, tudíž teoreticky umožňuje dosažení imerzní VR úrovně viz. (kap. three.js) .

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).

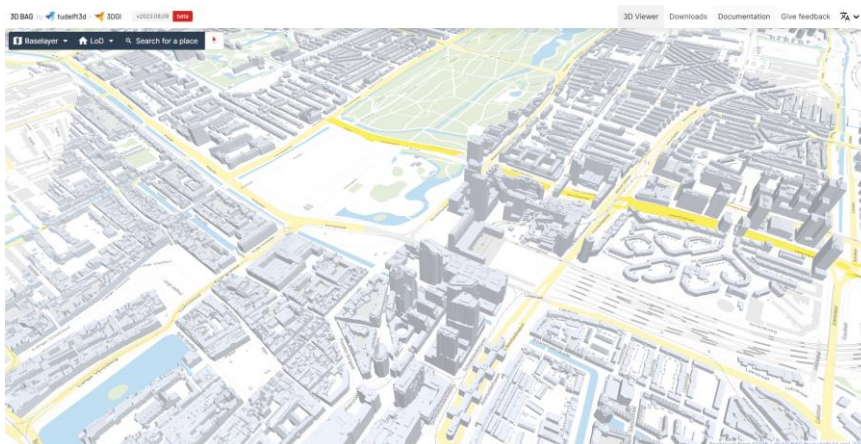
**CesiumJS** je knihovna umožňující tvorbu interaktivních 3D aplikací založených na geoprostorových datech. Zaměřená je převážně na zobrazení velkých datových sad jako jsou modely terénu, budovy a družicové snímky, a to i v globálním měřítku, zpravidla ve formátech 3D Tiles. Výhoda CesiumJS je její integrace 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, kdy poskytují 3D model budov, vegetace a terénu pro celé Švýcarsko.

**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 3D Tiles (budovy) a WMTS (podkladová mapa). Prohlížeč 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 aj. 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 záměrem 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 přiblížení a posun**. Knihovna momentálně nepodporuje WebXR API tudíž rozšíření míry imerze je momentálně nemožné a knihovna není vhodným kandidátem. (deck.gl 2023)



Obr. 23 3dbag-viewer. zdroj: (Peters et al. 2021)



Obr. 24 iTowns příklad zdroj: (iTown Contributors 2023)

Na základě výše uvedených řešení je možné určit, že pro geoprostorové informace dominuje především 3D Tiles formát a následně 3D renderery 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. Z této analýzy je možné vyvodit, že většina technologií je přímou implementací nad WebGL popř. využívá populární three.js knihovnu.

Tab. 5 Prozatímní tabulka technologií s parametry - mohou se změnit

S	WebGL	WebXR	Open Source	Dokumentace	Pracnost	Interoperabilita
ArcGIS + ArcGIS JS API	Ano	Ne	Ne	Ano	Střední	Ano
City Engine + VR Export 360°	Ano	Ano	Ne	Ano	Střední	Ne
QGIS + qgis2threejs	Ano	Možná	Ano	Ano	Vysoká	Ano
Luciad Ria	Ano	Ne	Ne	Ano	Vysoká	Ne

CesiumJS	Ano	Ne	Ano	ANo	Vysoká	Ano
mts-geospatial	Ano	Ne	Ano	Ano	Vysoká	Ano

#### 5.4.4 Webová řešení

##### A – Frame (HTML, Three.js)

Prototyp Petrov: <https://foam-jumpy-dianella.glitch.me>

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 jakékoliv 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.

Podpora pro 3D Tiles není dokonalá, pouze základní podpora pro three.js.

##### Mozilla Hubs + Spoke editor (Three, Aframe, WebRTC aj.)

Prototyp UrbanGrid: [https://hubs.mozilla.com/bBJ9sxc?hub\\_invite\\_id=Lr9efka](https://hubs.mozilla.com/bBJ9sxc?hub_invite_id=Lr9efka)

Prototyp 3D model Brno: <https://hubs.mozilla.com/jkemrr4>

Open source projekt vytvořen 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.

##### React Three Fiber (Three.js + ReactJS)

##### Needle engine (Unity, Three.js, WebXR)

Prototyp Petrov:

Viz: <https://interesting-parallel-bit.glitch.me>

kod: <https://glitch.com/edit/#!/interesting-parallel-bit>

Jedná se o webový runtime pro 3D aplikace. Umožňuje lokální vývoj, nasazení na jakýkoliv server, networking a WebXR rozhraní. Needle Exporter umožňuje propojení mezi Unity Editorem 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 využívá postupu, kdy je možné v binární formě (.glb) obsáhnout celou aplikaci nejen scénou. Primárním cílem Needle enginu je rychlá iterace při vývoji, responzivní design pro VR a AR, využití otevřených

Commented [JH12]: Opravené linky

standardů pro 3D a web, interoperabilita mezi nativními aplikacemi a webovými frameworky. (needle-tools 2023)

Momentálně (30.1.2023) je Needle Engine podporován pouze pro Unity Editor. Zde je vhodné porovnání s nativním Unity WebGL výstupem. Unity WebGL umožňuje export kompletních projektů 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. Needle engine se snaží o rychlou iteraci při vývoji a možnost takřka instantního prototypování při tvorbě Unity projektů v jejich webovém ekvivalentu. Needle engine tedy poskytuje možnost, jak propojit Unity Editor s klasickým přístupem k webovému vývoji. (needle-tools 2023)

Needle engine je nadstavbou nad Unity Editorem a three.js. 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 graf scény. Vlastní komponenty mohou být definovány pomocí .ts nebo .js a Needle Engine je následně automaticky přeloží do C# ekvivalentu, tudíž je možné s nimi automaticky pracovat v Unity. Needle Engine poskytuje funkcionalitu ve 3 hlavních formách (needle-tools 2023):

1. Souhrn předpřipravených komponentů a nástrojů, které umožňují tvorbu scény v rámci Unity Editoru
2. Export vytvořené scény do glTF formátu
3. Webové runtime prostředí, které načítá glTF soubory a vykresluje je pomocí three.js

#### 5.4.5 Webová řešení produkty

### 5.5 Praktické porovnání vybraných technologií

Metodika testování

Testování VR aplikace na webu by mělo projít 3 mi fázemi:

1. Zda-li je aplikace funkční při prohlížení na klasické obrazovce s klávesnicí a myší
2. Na emulátoru VR headsetu
3. Na mobilním telefonu
4. V HMD – Oculus Quest

#### 5.5.1 Výběr technologie

## 6 NÁVRH A IMPLEMENTACE VLASTNÍ APLIKACE

Jak otestovat zda je VR dobrá pro vizualizaci dat na webu?

Generalizovat co jsou to geografická data a následně ukázat základní kategorie ve VR.

Ve VR dává smysl ukazovat 3D data to je jasný. Dává ale smysl ukazovat i 2D data – jaká to jsou?

### GEOG DATA (brainstroming možností):

- Terén
  - 2d – vrstevnice, hypsometrie
    - Tech:
      - rastr - bitmapa (jpg)
        - zdroj: lokálně, služby – WMS, REST
      - vektor – features (GeoJSON, GML)
        - zdroj: lokálně, služby – WFS, REST
    - 3d – povrch
      - Tech:
        - Mesh: - tin (gltf, 3d tiles, i3s atd.)
          - Zdroj:
            - lokálně – gltf - nemůže být velké území – moc dat, popř. nějaký on demand loading – spatial subdivision - HLOD?
            - Tvorba:
              - Z height rastru
                - Subdivision surface v blenderu
                - Qgis to three.js stejný postup ale automaticky
              - služba – 3DTiles? - loaduje se jen to co se vidí - cesium, vts-geospatial,
                - Google Maps 3DTiles API – jak získat data od Google Maps API lokálně??
                  - Textured 3D mesh
              - Cesium 3D
      - Objekty – budovy, silnice, vegetace, jednotlivé objekty
        - 2d – body, linie, polygony
          - Tech:
            - rastr - bitmapa (jpg)

- zdroj:
      - lokálně – tif, jpg atd.
      - služby – WMS, REST
  - vektor – features (GeoJSON, GML)
    - zdroj:
      - lokálně – shapefile, GeoJSON aj.
      - služby – WFS, REST
- 3d – instancované objekty
  - Tech:
    - Mesh: – 3d modely – tin – (glTF, cityjson, cityGML, obj, collada atd.)
      - Zdroj:
        - Lokálně – data naložovaná do klienta při otevření aplikace
        - Služby - ?? – cdn? – sketchfab – vlastní server serving?
    - Mesh – jednoduchý – plocha s texturou co se otáčí – symbol (format???)
      - Zdroj:
        - ??
- Text
  - ...
- Tematická data – statistika, agregace, atd
  - 2d – body, linie, polygony
    - Tech:
      - rastr - bitmapa (jpg)
        - zdroj:
          - lokálně – tif, jpg atd.
          - služby – WMS, REST
      - vektor – features (GeoJSON, GML)
        - zdroj:
          - lokálně – shapefile, GeoJSON aj.
          - služby – WFS, REST
  - 3d – volumetrické
    - Tech:
      - Mesh:



- stejné
- Voxely
  - ??

Odkud data?

- ČR – dmr, dmp,
  - Brno – všechno
- Švýcarsko – swiss topo
- Google API
  - <https://developers.google.com/maps/documentation/tile/use-renderer>

## 6.1 Definice uživatelských požadavků

Za účelem úspěšné implementace je nutné definovat směr jakým by se aplikace měla ubírat. Je nutné zpočátku zmínit, že není v zájmu této práce vyvinout robustní univerzální VR aplikaci pro vizualizaci geografických dat, a to primárně z toho důvodu, že se jedná o komplexní problém, na němž pracují společnosti se značně většími znalostními, časovými a finančními zdroji. Je tedy důležité pečlivě vybrat některé aspekty tvorby takovéto aplikace a zaměřit se na ně.

Aplikace by měla primárně vizualizovat geografická data. Ačkoliv jak z definice VR vyplívá určitá míra interakce by měla být možná. Minimální požadavky na interakci by měli být pohyb uživatele, rozšířeným požadavkem pak interakce se samotnými daty. Aplikace by měla jasně sdělovat geografickou polohu dat v geoprostorovém kontextu.

Mezi definovanými požadavky je nutné vytvořit hierarchii dle priority požadavků. Jednou z metodik využívaných v softwarovém inženýrství je metoda MoSCoW. Jedná se o skupinu zkratk pro:

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

Pomocí této metody je možné konkretizovat obecně zvolené požadavky při výběru technologie na konkrétní aplikaci.

### 6.1.1 Poučení z existujících řešení

**OSM a Aframe** - (VR Map 2019)

- World is flat
  - No curvature
  - No terrain

- Ground tiles change size based on lon lat, in AFRAME coord sys je v metrech – tiles z vlastního cache serveru
- Budovy a stromy z Overpass API
- Kamera a Kontroler setup – vlastní modifikace (létání atd)
- Code
  - Html - Start dialog popup, scene, camera, controllers
  - Map.js – load, handlers, fetch from Overpass API
  - Conversions – coordinate conversions
  - Position-limit.js – aframe component to keep position above ground
  - Tiles, trees.js, buildings.js – draw objects to the screen

#### Funkční požadavky:

##### Orientace

- Aplikace by měla podporovat orientaci polohy v rámci scény – orientace virtuální kamery, poloha v rámci celé scény (přehledová mapa)

##### Zobrazení geografických dat 3D i 2D

- a. Terén, Plošné (budovy, objemy – tematická kart.), liniové (stuhová mapa), bodové (stromy)
- b. Podklad (textury terénu – ortofoto, 2D mapy), Textury objektů

##### 2. Možnost interakce

- a. DOFs
- b. Interakce s objekty – atributové info, přepínání vrstev (nejde v mozila hubs), pohyb s objekty, analýzy

##### Nefunkční požadavky

1. Použitelnost – Aplikace by měla být pro uživatele intuitivní, v případě virtuální reality se jedná především o navigaci ve 3D prostředí, zároveň se jedná o intuitivnost doplňkového uživatelského rozhraní
2. Dostupnost – Aplikace by měla být dostupná 99% času.
3. Výkonost – Aplikace by měla být dostatečně výkonná tak aby zamezila nízkým hodnotám vykreslování, které mohou vést k značnému snížení imerze.

## 6.2 Výběr HMD

HMD na pro které bude vizualizace primárně implementována a následně testována.

1. Low-Cost – Android 10 – Mi A2 Lite - Retrak Utopia 360° VR Headset - doma
2. Střední – Oculus Quest 2 – škola/doma

3. Hight end – HTC Vive - škola

#### 6.2.1 Prohlížeče

1. Oculus Browser
2. Chrome
3. Firefox
4. Samsung internet
5. Microsoft Edge
6. Wolvic (*Firefox reality*)

Optimalizace – Filtrace - (Coltekin, Reichenbacher 2011) – progresive loading – compresion

### 6.3 Metodika

- Viz (Sherman, Craig 2019)

#### 6.3.1 Návrh

(Coltekin et al. 2020) dělí přístupy k návrhu XR na *visualisation design* a *interaction design*.

1. Před příprava dat – výběr, konverze do vhodného formátu, oprava chyb, optimalizace pro web (snížení podrobnosti, optimalizace textur, optimalizace velikosti).
2. Publikace dat na web – Kde to bude hostnutý ty data?
3. Tvorba webové stránky s vizualizací dat
4. Hostování webové stránky na webu

### 6.4 UX

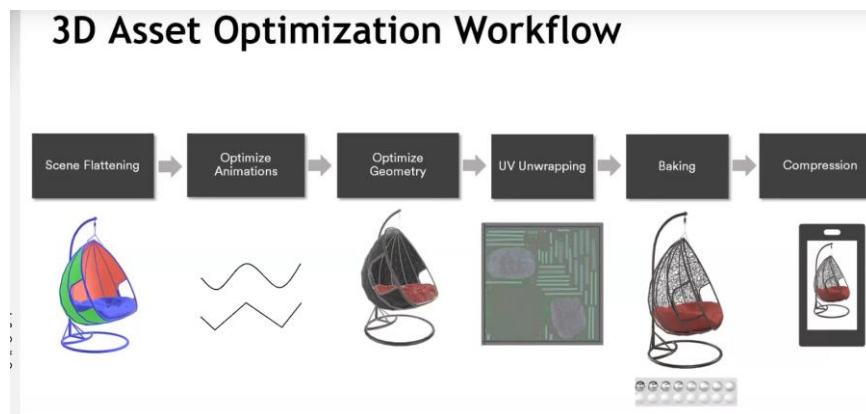
Menu -

## **7 UŽIVATELSKÉ TESTOVÁNÍ**

## 8 DISKUZE

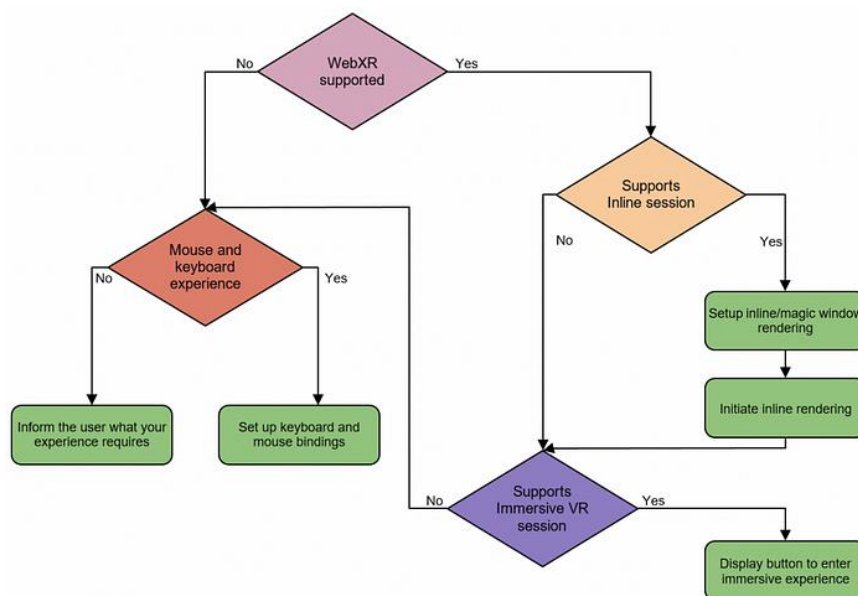
Optimalizace Paměti a Výkonu

### 3D Asset Optimization Workflow



Tech Stacky se mění ale koncepty a technologické přístupy jsou obdobné. Výběr technologie tedy není hlavním cílem, jako popis a analýza způsobu vizualizace dat.

Acesibility



## 9 ZÁVĚR

## **PŘÍLOHY**

## **BIBLIOGRAFIE**



## **SEZNAM PŘÍLOH**

Příloha č

