

**Technická univerzita v Košiciach  
Fakulta elektrotechniky a informatiky**

**Transformácia objektov reálneho sveta do  
virtuálnych prostredí**

**Diplomová práca**

**2022**

**Bc. Michal Bardzák**

**Technická univerzita v Košiciach  
Fakulta elektrotechniky a informatiky**

**Transformácia objektov reálneho sveta do  
virtuálnych prostredí**

**Diplomová práca**

Študijný program: Informatika  
Študijný odbor: 9.2.1. Informatika  
Školiace pracovisko: Katedra počítačov a informatiky (KPI)  
Školiteľ: Ing. Štefan Korečko, PhD.  
Konzultant:

**Košice 2022**

**Bc. Michal Bardzák**

## **Abstrakt v SJ**

Hlavnou náplňou tejto diplomovej práce je snaha preskúmať možnosti transformácie objektov z reálneho sveta do virtuálneho tak aby bol výsledok použiteľný pre webovú virtuálnu realitu. Toho je dosiahnuté použitím voľne dostupných prostriedkov na vytvorenie časti mesta Košice v niekoľkých úrovniach zložitosti.

## **Kľúčové slová v SJ**

Webová virtuálna realita, webová rozšírená realita, transformácia

## **Abstrakt v AJ**

The main purpose of this master thesis is explore possibilities of transforming real world objects into virtual in a way that makes it usable for web based virtual reality. This is achieved by using freely accessible resources to create part of city of Košice in several levels of complexity.

## **Kľúčové slová v AJ**

web virtual reality, web extended reality, transformation

## **Bibliografická citácia**

BARDZÁK, Michal. *Transformácia objektov reálneho sveta do virtuálnych prostredí*. Košice: Technická univerzita v Košiciach, Fakulta elektrotechniky a informatiky, 2022. 36s. Vedúci práce: Ing. Štefan Korečko, PhD.

**TECHNICKÁ UNIVERZITA V KOŠICIACH**  
**FAKULTA ELEKTROTECHNIKY A INFORMATIKY**  
Katedra počítačov a informatiky

**Z A D A N I E**  
**D I P L O M O V E J   P R Á C E**

Študijný odbor:     **Informatika**  
Študijný program:   **Informatika**

Názov práce:

**Transformácia objektov reálneho sveta do virtuálnych prostredí**  
Real World Objects Transformation to Virtual Environments

Študent:                 **Bc. Michal Bardzák**

Školiteľ:                **Ing. Štefan Korečko, PhD.**

Školiace pracovisko:   **Katedra počítačov a informatiky**

Konzultant práce:

Pracovisko konzultanta:

Pokyny na vypracovanie diplomovej práce:

1. Analyzovať súčasné prístupy k transformácii objektov reálneho sveta do virtuálnych prostredí.
2. Pri analýze sa zamerať na riešenia voľne dostupné a použiteľné pre webovú rozšírenú realitu (Web XR).
3. Navrhnuť a opísať konkrétne postupy takejto transformácie pre Web XR.
4. Navrhnuť a implementovať sadu virtuálnych prostredí pre overenie vplyvu zložitosti transformovaných objektov na výkon.
5. Realizovať overenie vplyvu zložitosti transformovaných objektov na výkon vybraných zariadení.
6. Vypracovať dokumentáciu podľa pokynov vedúceho práce.

Jazyk, v ktorom sa práca vypracuje: slovenský

Termín pre odovzdanie práce: 31.08.2022

Dátum zadania diplomovej práce: 29.10.2021



A handwritten signature in blue ink, likely belonging to prof. Ing. Liberios Vokorokos, PhD.

prof. Ing. Liberios Vokorokos, PhD.

dekan fakulty



## **Čestné vyhlásenie**

Vyhlasujem, že som záverečnú prácu vypracoval(a) samostatne s použitím uvedenej odbornej literatúry.

Košice, 31.8.2022

.....  
*Vlastnoručný podpis*

## **Podakovanie**

Na tomto mieste by som rád poďakoval svojmu vedúcemu práce za jeho čas a odborné vedenie počas riešenia mojej záverečnej práce.

Rovnako by som sa rád poďakoval svojim rodičom a priateľom za ich podporu a povzbudzovanie počas celého môjho štúdia.

# Obsah

---

<b>Úvod</b>	<b>1</b>
<b>1 Analytická časť</b>	<b>2</b>
1.1 Rekonštrukcia využitím stereo-fotogrametrie . . . . .	3
1.2 Využitie mapových údajov . . . . .	5
1.3 Rekonštrukcia objektu pomocou mračna bodov . . . . .	6
1.4 3D Modelovací program Blender . . . . .	7
1.5 Popis doplnku Blender-OSM . . . . .	8
1.6 Popis knižnice Buildify . . . . .	9
1.7 Virtuálna Realita . . . . .	10
1.8 Validácia softvérových prototypov vo virtuálnej realite . . . . .	11
1.9 Webová virtuálna realita . . . . .	12
1.10 Popis rámca webovej virtuálnej reality A-Frame . . . . .	12
<b>2 Syntetická časť</b>	<b>14</b>
2.1 Návrh a implementácia sady virtuálnych prostredí . . . . .	14
2.2 Vytváranie scény vo webovej virtuálnej realite . . . . .	15
2.3 Návrh a implementácia skriptov pre generovanie A-Frame scény . .	21
2.4 Kolízie vo webovom rámci A-Frame . . . . .	24
2.4.1 Problémy s kolíziami . . . . .	25
2.5 Importovanie budov do scény . . . . .	27
2.5.1 Problémy s výkonom pri vyššom počte budov . . . . .	28
2.6 Pohyb používateľa po scéne . . . . .	28
2.7 Stromy v A-Frame scéne . . . . .	29
2.7.1 Rotácia dvojrozmerných billboardov . . . . .	30
2.7.2 Používanie inštancií 3D modelov stromov . . . . .	30
<b>3 Meranie výkonu vytvorených scén</b>	<b>32</b>
<b>4 Záver</b>	<b>34</b>

<b>Literatúra</b>	<b>35</b>
<b>Zoznam príloh</b>	<b>37</b>

# Zoznam obrázkov

---

1.1	Skalka v záhradke zrekonštruovaná pomocou programu Meshroom.	4
1.2	Plynomer v tvare domčeka zrekonštruovaný pomocou programu Meshroom. . . . .	4
1.3	Časť mesta Košice vytvorená použitím mapových údajov. . . . .	5
1.4	Lod získaná rekonštrukciou z mračna bodov v programe MeshLab.	6
1.5	Počiatočný stav programu Blender. . . . .	7
1.6	Knižnica Buildify v kombinácii s doplnkom Blender-OSM. . . . .	10
1.7	Hello world, respektíve Hello WebVR v rámci A-Frame. . . . .	13
2.1	Vzhľad jednotlivých úrovní, konkrétne úrovne 1, 4, 7. . . . .	16
2.2	Šípku v pravom hornom rohu . . . . .	17
2.3	Webová stránka s mapou sveta. . . . .	18
2.4	Doplnok vytvorený pre využitie v tejto práci. . . . .	20
2.5	Budovy pred a po predĺžení. . . . .	22
2.6	Budovy s textúrami a farbami. . . . .	23
2.7	Budovy zo správnymi a nesprávnymi kolíziami. . . . .	26
2.8	Dvojrozmerný strom. . . . .	30
2.9	Trojrozmerný strom. . . . .	31
3.1	Tabuľka a graf vyhodnotenia meraní. . . . .	33

# Úvod

---

Problémom riešeným v tejto práci je transformovanie objektov reálneho sveta na objekty použiteľné vo virtuálnych prostrediach. Táto téma je lákavá hlavne z toho dôvodu že takáto transformácia objektov môže značne urýchliť tvorbu virtuálnych zážitkov a to hlavne v porovnaní s tradičnými metódami ako je napríklad 3D modelovanie. Zatiaľ čo existuje množstvo takýchto metód zaoberajúcich sa transformáciou objektov z reálneho sveta do virtuálneho, žiadna z nich nieje perfektná a každá má svoje výhody a nevýhody. Z tohto dôvodu existuje veľa rôznych prác, ktoré sa touto problematikou zaoberajú a snažia sa zlepšiť už existujúce metódy, ktoré sú momentálne používané alebo vytvoriť nové metódy, ktoré by potenciálne mohli nahradiť tie už zaužívané. Pri týchto metódach je možné sa zamerať na rôzne aspekty ktoré môžu byť zlepšené ako napríklad kvalita objektov vytvorených týmito metódami, škála na ktorej sú tieto metódy aplikovateľné, ich efektívnosť, časová náročnosť a podobne. Je taktiež potrebné zobrať v úvahu že každá z týchto metód má nielen svoje klady a zápory ale aj špecifické využitie ktoré ostatné metódy nemusia vôbec ponúkať. Preto je nutné preskúmať rôzne metódy transformácie reálnych objektov do prostredí virtuálnej reality a nájsť tie, ktoré sú vhodné na účeli tejto práce.

## Formulácia úlohy

Cieľom v tejto práci bude metódy transformácie objektov reálneho sveta do virtuálnych prostredí preskúmať a následne sa zamerať na tie ktoré sú využiteľné v prostredí webovej rozšírenej reality. Po tom čo budú tieto metódy preskúmané bude potrebné overiť či sú naozaj použiteľné vo virtuálnych prostrediach poskytovanými webovou rozšírenou realitou. Na overenie týchto riešení bude potrebné navrhnuť a popísať postup takejto metódy a vykonať experimenty a merania ktoré potvrdia alebo vyvrátia jej funkčnosť.

# 1 Analytická časť

---

V súčasnosti existuje veľké množstvo rôznych prác zaoberajúcich sa transformáciou objektov reálneho sveta do virtuálnych prostredí. Príkladmi takýchto prác je napríklad článok [1] kde je popísaný automatizovaný prístup k takejto transformácii objektov. Tento postup bol implementovaný ako sada algoritmov a senzorov v špecializovanom zariadení na trojdimenzionálnu rekonštrukciu nazývanom EST (Environmental Sensor for Telepresence). Ďalším príkladom takejto práce je článok [2], v ktorom je popísaný prístup k rekonštrukcii vonkajších prostredí na veľkej škále, využitím mobilného zariadenia. Mobilne zariadenie použité bolo Project Tango čo je experimentálny tablet od firmy Google. Ako je však zrejmé, veľká časť z týchto postupov je závislá na používaní špecializovaných nástrojov, ktoré môžu stáť podstatné množstvo finančných prostriedkov alebo nemusia byť dostupné pre verejnosť vôbec. V dnešnej dobe sa však na mobilný trh dostáva stále viac a viac zariadení s technológiami ktoré umožňujú vykonávať skenovanie reálnych objektov. Výkon osobných počítačov ako aj softvér, ktorý umožňuje transformáciu objektov reálneho sveta, a jeho možnosti sa taktiež zlepšujú. Užívatelia osobných počítačov majú možnosť vykonávať transformáciu objektov z reálneho sveta pomocou stereo-fotogrametrie, rekonštrukcie objektov z mračien bodov, používanie mapových údajov a v budúcnosti dokonca aj zatiaľ experimentálnych metód ako napríklad NeRF, ktorá bola použitá v práci [3], na rekonštrukciu oblasti Alamo Square v americkom meste San Francisco. Na rekonštrukciu bolo použitých 2,8 milióna fotografií, avšak oblasť, ktorá bola rekonštruovaná je približne jeden kilometer štvorcový. Ďalšia z takýchto experimentálnych metód je opísaná v práci [4], ktorá ponúka rekonštrukciu vyššej kvality a s vyššou rýchlosťou, než momentálne najmodernejšie používané metódy. Vďaka týmto technológiám sa vytváranie realistických virtuálnych prostredí a objektov, na základe ich náprotivkov zo skutočného sveta, stáva čoraz dostupnejšie a jednoduchšie pre verejnosť ako aj profesionálov pracujúcich v oblasti počítačovej grafiky.

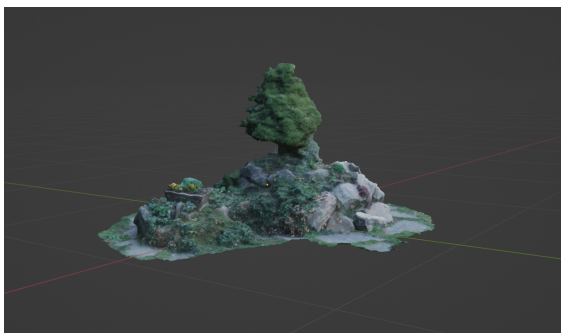
## 1.1 Rekonštrukcia využitím stereo-fotogrametrie

Stereo-Fotogrametria je veda zaoberajúca sa získavaním kvantitatívnych dát trojrozmerných objektov z fotografií. Fotografia je projekcia trojrozmerného prostredia do dvojrozmernej formy, pričom je stratená hĺbková informácia.

Stereo-Fotogrametria je snaha zvrátiť tento proces a vytvoriť trojrozmerný objekt z informácii dostupných len v dvojrozmernom formáte. V najčastejšom prípade ide konkrétne o digitálnu stereo-fotogrametriu. Kniha [5] uvádza že digitálna stereo-fotogrametria má väzby na rozpoznávanie vzorov, čo je súčasťou strojového videnia.

Na vytvorenie objektov, ktoré by bolo možné použiť vo webovom rámci pre virtuálnu realitu A-Frame bola vyskúšaná digitálna stereo-fotogrametria. Konkrétne bol použitý slobodný a open-source program s názvom Meshroom. Článok [6] opisuje Meshroom ako program na rekonštrukciu 3D pomocou fotogrametrického strojového videnia založeného na rámci AliceVision. Tento program ponúka prostredie na vykonávanie rôznych úkonov softvérového videnia. Do programu je potrebné nahráť fotografie objektu ktorého trojrozmerný model chceme získať a po kliknutí tlačidla Start, program automaticky vytvorí model objektu z poskytnutých fotografií. Fotografie musia byť v dostatočne vysokom rozlíšení a nesmú byť rozmazané, vzhľadom na to, že to môže výrazne ovplyvniť kvalitu výsledného modelu či dokonca spôsobiť že model, ktorý sa snaží program Meshroom pomocou stereo-fotogrametrie zrekonštruovať, nebude vôbec zostrojený. Existujú taktiež obmedzenia na objekty, na ktoré môže byť táto metóda aplikovaná. Objekt nemôže byť príliš tenký, lesklý alebo priehľadný. Taktiež je kvôli kvalite výslednej textúry potrebné aby bol objekt pod rozptýlením alebo rovnomerným svetlom, ako napríklad pri zamračenom počasí. Ak je táto podmienka splnená, nebude vznikať príliš veľký rozdiel v tmavosti tieňov. Rekonštrukcia modelu môže zabrať niekoľko hodín v závislosti od počtu a rozlíšenia fotografií, a taktiež výkonu počítača na ktorom je vykonávaná. Na počítači, na ktorom bol test tohto programu vykonávaný to zabralo približne štyri hodiny pri päťdesiatich-piatich fotografiách s rozlíšením 4000 x 3000.





Obr. 1.1: Skalka v záhradke zrekonštruovaná pomocou programu Meshroom.

Vzhľadom na to, že riešenie má byť použiteľné pre webovú virtuálnu realitu bol vo výslednom modeli taktiež znížený počet vrcholov na stotisíc, rozlíšenie textúr bolo upravené tak aby nezaberali príliš veľké množstvo priestoru a sú tak približne 8x menšie ako tie pôvodné.



Obr. 1.2: Plynomer v tvare domčeka zrekonštruovaný pomocou programu Meshroom.

Program Meshroom však nedokáže izolovať objekt ktorý chceme pomocou stereo-fotogrametrie získať a preto je pri rekonštrukcii vytvorené aj okolie objektu na fotografiách. Z tohto dôvodu je nutné odstrániť všetky nechcené časti trojrozmerného modelu. Na tento účel je možné použiť nástroj s názvom MeshLab. Meshlab je open-source program ponúkajúci sadu nástrojov na úpravu, čistenie, inšpekciu, renderovanie, textúrovanie a konverziu trojrozmerných modelov. Po odstránení artefaktov a prebytočných častí modelu, je model možné exportovať z programu MeshLab a importovať ho do programu Blender, v ktorom je možné odstrániť prebytočné vrcholy, ktoré neboli pripojené k žiadnej hrane a vrcholy ktoré sa na-

vzájom prekrývali. Následne program Blender ponúka možnosť exportovať model vo formáte gltf, čo je štandardný formát pre trojdimenzionálne scény a modely podporovaný a vyvíjaný Khronos Group, ktorý zabezpečuje minimalizovanie veľkosti súboru. Tento súbor je možné používať priamo vo webovom rámci A-Frame.

## 1.2 Využitie mapových údajov

Pri skúmaní metód na vytváranie realistických prostredí bola taktiež preskúmaná možnosť vytvárania realistického modelu mesta, využitím mapových údajov voľne dostupných na internete. Podobný nápad bol preskúmaný v práci [7], kde bola poskytnutá metodológia na generovanie nízko rozpočtových 3D máp využitím voľne dostupných, dvojrozmerných údajov budov a výškových údajov získaných zo satelitných dátových sád. Tu sa používanie mapových údajov z OpenStreet-Map na vytváranie trojrozmerného modelu mesta ukázalo ako schopná alternatíva komerčných modelov.

V článku [8] je OpenStreetMap projekt opísaný ako kolektív ktorý poskytuje používateľmi generované mapy a pracuje na podobnom modeli ako wikipédia. Cieľom projektu je vytvoriť súpravu mapových údajov ktoré sú voľne použiteľné a upravovateľné. Podstatné množstvo prispievateľov kolaboratívne upravuje svetovú mapu použitím OpenStreetMap technickej infraštruktúry zatiaľ čo skupina dobrovoľníkov, ktorá je jadrom tohto projektu, pracuje na zlepšovaní tejto infraštruktúry.

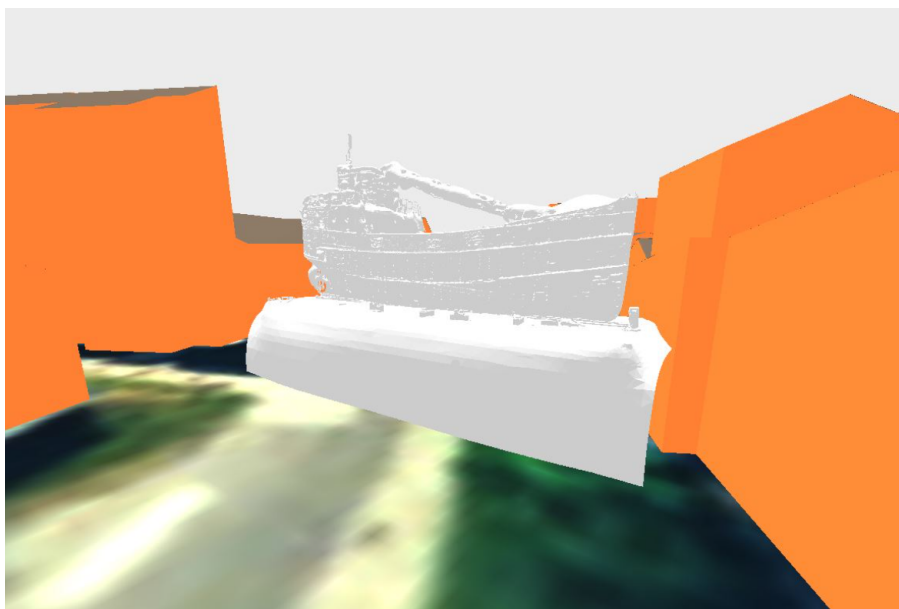


Obr. 1.3: Časť mesta Košice vytvorená použitím mapových údajov.

### 1.3 Rekonštrukcia objektu pomocou mračna bodov

Jednou z možností pre transformáciu objektov reálneho sveta do virtuálnych prostredí, je rekonštrukcia trojrozmerných objektov z mračien bodov. Rozdiel medzi týmto prístupom a stereo-fotogrametriou spočíva v tom, že pri rekonštrukcii z mračna bodov máme k dispozícii aj hĺbkové údaje, zatiaľ čo pri stereo-fotogrametrii tieto údaje chýbajú. Oproti stereo-fotogrametrii má taktiež tú výhodu že je rekonštrukciu z mračna bodov možné použiť aj pri transformácii väčších objektov ako napríklad budov alebo terénu. Vo vedeckom článku [9] je ako jedným z najčastejších prístupov k tomuto problému opísané odhadnutie normálového vektora každého bodu reprezentujúceho povrch objektu, a následná rekonštrukcia trojrozmerného modelu z týchto odhadnutých normálových vektorov. Na tento úkon je možné použiť niekoľko rôznych nástrojov, z ktorých sú niektoré voľne dostupné k stiahnutiu.

Pri testovaní transformácie mračna bodov na trojrozmerný objekt bol použitý open-source softvér s názvom MeshLab. MeshLab umožňuje vykonať výpočet normálového vektora pre každý bod v mračne bodov, rekonštrukciu vodotesného povrchu trojrozmerného objektu z vypočítaných normálových vektorov a taktiež zníženie počtu hrán modelu, ktorý bol rekonštrukciou získaný, využitím zjednotenia hrán daného modelu.



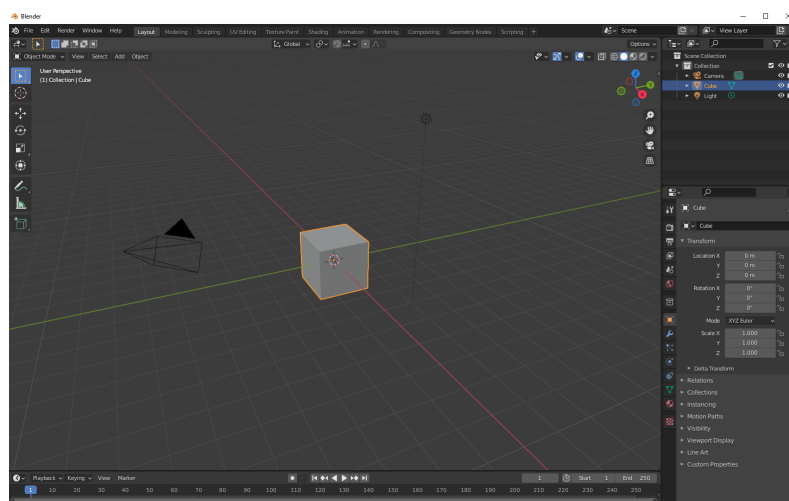
Obr. 1.4: Lod získaná rekonštrukciou z mračna bodov v programe MeshLab.

Aj napriek tomu že má táto metóda veľký počet výhod oproti tradičnej stereo-fotogrametrii, jej veľkou nevýhodou je nutnosť použitia špecializovaného vyba-

venia používaného na skenovanie trojdimenzionálnych objektov. V dnešnej dobe sa však dostávajú technológie ako LiDAR, ktoré sú používané na skenovanie, aj na mobilné telefóny. To znamená že v blízkej budúcnosti už špecializované vybavenie bude pravdepodobne nutné iba pre profesionálov v tomto odvetví.

## 1.4 3D Modelovací program Blender

Blender je slobodný a open-source softvér ktorý sa hlavne zameriava na vytváranie a prácu s 3D projektmi rôzneho druhu. Blender je taktiež multiplatformový softvér, vďaka čomu podporuje a funguje rovnako dobre na rôznych platformách ako napríklad Microsoft Windows, GNU/Linux, Apple Macintosh a podobne. Podporuje zrefazované spracovanie 3D, vrátane modelovania, animácie, simulácie, renderovania, obrazovej kompozície a snímania pohybu. Blender taktiež podporuje aj vedľajšie funkcie ako editovanie videí a vytváranie textúr. Pokročilí používatelia sú schopní v programe Blender taktiež používať API (Application Programming Interface v preklade rozhranie pre programovanie aplikácii), na vytváranie skriptov ktoré slúžia na prispôsobenie aplikácie a vytváranie špecializovaných nástrojov. Pri písaní týchto skriptov sa používa programovací jazyk Python. Blender je vhodný pre individuálnych tvorcov a menšie štúdiá ktorým prináša úžitok zjednotené zrefazované spracovanie a prístupné vývojové prostredie.



Obr. 1.5: Počiatočný stav programu Blender.

## 1.5 Popis doplnku Blender-OSM

Blender-OSM je doplnok pre 3D modelovací softvér Blender, ktorý poskytuje možnosť jedným klikom stiahnuť a importovať údaje z OpenStreetMap a údaje o teréne s celosvetovým pokrytím. Doplnok je dostupný zadarmo a zdrojový kód k doplnku je dostupný pod GPL licenciou. Pomocou tohto doplnku je možné stiahnuť a importovať niekoľko rôznych druhov objektov ako sú napríklad budovy, vodné telesá, oblasti obsahujúce vegetáciu, cesty, chodníky, koľajnice a terén. Budovy majú okrem pôdorysu a súradníc dostupné aj údaje ako výška a počet poschodí. Tie sú používané pri vytváraní finálnej scény. Majú aj niekoľko rôznych typov striech z ktorých si je možné vybrať pri importovaní budov. Ak je terén dodaný, budovy budú položené na terén automaticky. Vodné telesá sú importované ako polygóny a ak je terén prítomný, importované vodné telesá sú na neho vykreslené. Oblasti obsahujúce vegetáciu sú podobne ako vodné telesá importované ako polygóny a v prípade prítomnosti terénu, sú na objekt terénu vykreslené. Importované cesty, chodníky a koľajnice majú šírku a sú v programe Blender reprezentované ako krivky. Pri importovaní terénu, stiahne a importuje doplnok Blender-OSM údaje digitálneho výškového modelu vo formáte SRTM. Údaje sú pripravené spoločnosťou Mapzen z voľne dostupných súborov dát ktoré sú následne dostupné na Amazon Web Services. Hustota rozostupu vzorky meraných bodov v údajoch ktoré sú dostupné je jedna uhlová sekunda alebo približne tridsať metrov na zemepisnej šírke a zemepisnej dĺžke.

Pri inštalácii doplnku je odporúčané používať najnovšiu stabilnú verziu programu Blender. Doplnok je možné stiahnuť z webovej stránky <https://prochitecture.gumroad.com/l/blender-osm> kde je dostupný zadarmo. Súbor je stiahnutý vo formáte zip s názvom `blosm.zip`. V editore programu Blender je nutné kliknúť na položku hlavného menu Edit. Po kliknutí sa otvorí rozotváracia ponuka kde je potrebné kliknúť na tlačidlo Preferences. V okne, ktoré sa otvorí, sa na ľavej strane nachádza menu kde je potrebné stlačiť tlačidlo Add-ons (Po slovensky doplnky). Ďalej je nutné kliknúť na tlačidlo Install ktoré otvorí okno na výber súborov. V tomto okne je potrebné vyhľadať a vybrať stiahnutý súbor s názvom `blosm.zip` a nainštalovať ho pomocou tlačidla Install. Potom čo bol doplnok nainštalovaný je nutné ho v záložke Add-ons nájsť a aktivovať zaškrtnutím políčka. Ďalším krokom je zadať cestu k priečinku do ktorého majú byť sťahované údaje z OpenStreetMap a vložiť Mapbox Access Token. V prípade ak používateľ nemá Mapbox Access Token je možné ho získať na webovej stránke ktorá sa otvorí

po kliknutí na tlačidlo Get It!. Ak používateľ nemá na tejto webovej stránke účet je možné                      sa                      zadarmo                      zaregistrovať.

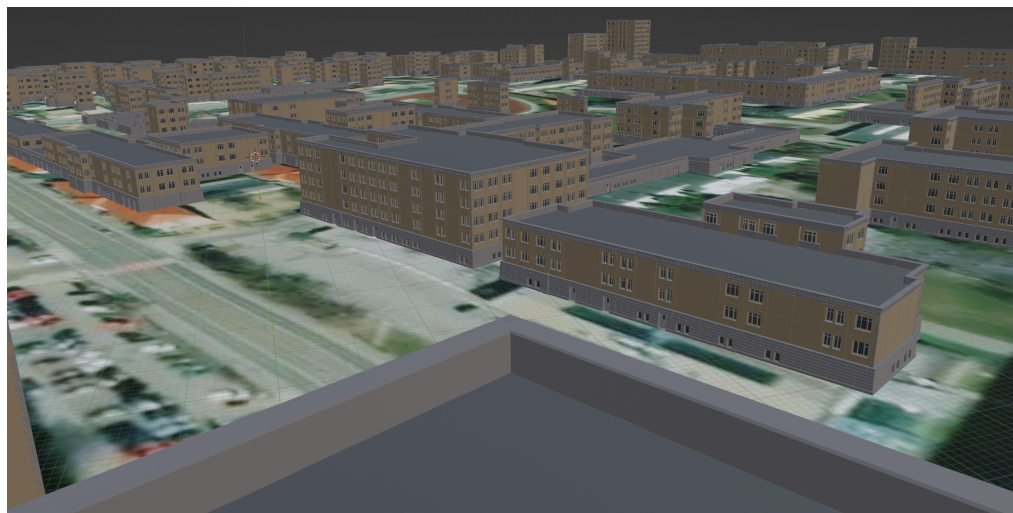
## 1.6 Popis knižnice Buildify

Buildify je knižnica geometrických uzlov pre jednoduché vytváranie budov v 3D modelovacom programe Blender. Využíva modulárne modely budov, ktoré môžu byť použité pre vytvorenie realistických scén, v ktorých je potrebné mať rôzne vyzerajúce budovy. Táto knižnica je kompatibilná s doplnkom Blender-OSM ktorý dokáže vygenerovať geometrické rozloženie základov budov a taktiež dokáže nastaviť niekoľko parametrov používaných geometrickými uzlami.

Pri testovaní knižnice Buildify s doplnkom Blender-OSM však bolo zistené, že pri veľkosti scény vytvorenej v tejto práci, počítač na ktorom bol tento pokus robený nemal dostatočné zdroje na vytvorenie scény s knižnicou Buildify. Tento problém by pravdepodobne bolo možné riešiť napríklad importovaním vlastných modulárnych modelov budov s nižšou zložitou než tie, ktoré prichádzajú priložené v knižnici. To však nebolo otestované vzhľadom na nedostatok času a skúseností v 3D modelovaní. Je tu taktiež prítomný problém spôsobovaný doplnkom Blender-OSM, kde geometrické rozloženie základov budov je importované ako jeden celok na ploche, kvôli čomu budovy vygenerované knižnicou Buildify nekopírujú terén ale nachádzajú sa v jednej rovine. Je možné že je to len problém spôsobený nezrelosťou knižnice Buildify a jej kompatibilitou s doplnkom Blender-OSM, ktorý bude opravený v budúcnosti.

Značne zjednodušená scéna ktorú sa podarilo exportovať pomocou knižnice Buildify však stále mala viac ako 400 megabajtov, kvôli čomu nie je veľmi vhodná na využitie vo webovej virtuálnej realite.





Obr. 1.6: Knižnica Buildify v kombinácii s doplnkom Blender-OSM.

## 1.7 Virtuálna Realita

Virtuálna realita odkazuje na virtuálne prostredia ktoré môžu ale nemusia napodobňovať skutočný svet a ktoré je možné používať vo veľkom množstve odvetví na veľké množstvo účelov. V súčasnosti sa virtuálna realita používa v oblastiach ako zábavný priemysel, kde je táto technológia často používaná pri počítačových hrách alebo v odvetví virtuálneho turizmu, vďaka čomu môžu používatelia virtuálnej reality navštíviť miesta, na ktoré by sa za normálnych okolností nedostali, priamo z pohodlia domova. Virtuálnu realitu je taktiež možné používať pri školení zamestnancov, ktorí pracujú v odvetví s vysokým rizikom ako to bolo opísané v článku [10], kde bola skúmaná možnosť využitia virtuálnej reality pri bezpečnostnom školení stavebných pracovníkov, a kde sa ukázalo v istých úkonoch školenie vo virtuálnej realite efektívnejšie pri udržiavaní pozornosti a sústredenia účastníkov tohoto výskumu oproti tradičným metódam bezpečnostného školenia.

V knihe [11] autor pojednáva o tom že človek sa snažil vytvoriť vierohodné ilúzie už stáročia avšak až digitálne počítače umožnili vznik konceptu virtuálnej reality ako ju poznáme dnes. Vďaka moderným počítačom môže virtuálna realita byť interaktívna a pri kombinácii s robotikou môže pozostávať aj z iných komponentov ako len vizuálnych a akustických.

V neposlednom rade je možné použiť virtuálnu realitu pri tréňovaní modelov strojového učenia. Tréňovanie modelov strojového učenia vo virtuálnej realite má

obrovské výhody oproti vykonávaniu tréningu takýchto modelov v reálnom svete. Jedným a pravdepodobne najdôležitejším dôvodom na využitie virtuálnej reality je cena. Či už z hľadiska peňažných prostriedkov alebo času stráveného tréningom modelov strojového učenia. Čo sa týka výhody nižších nákladov na tréning modelov strojového učenia vo virtuálnej realite, je dobrým príkladom tréning autonómnych vozidiel. Tréning modelu strojového učenia na riadenie autonómneho vozidla vo virtuálnej realite je oveľa lacnejšie než tréning v reálnom svete z dôvodov ako nehodovosť alebo cena paliva reálnych vozidiel. V prípade že riadiaci model zlyhá vo virtuálnej realite tak nenastane žiadny problém, zatiaľ čo v reálnom svete by následky mohli byť fatálne. V článku [12] je opísané využívanie virtuálneho prostredia vo videohre na tréning a testovanie autonómneho jazdného modelu. Čo sa týka časovej ceny tréningu je možné vykonávať strojové učenie vo virtuálnej realite vo väčších kvantitách než v reálnom svete a taktiež so zrýchleným časom vo virtuálnom prostredí oproti reálnemu času ktorý ubehol počas tréningu. Táto vlastnosť tréningu modelov vo virtuálnom prostredí bola preskúmaná v článku [13], kde umelá inteligencia vykonala desať rokov strojového učenia, zatiaľ čo v skutočnom čase ubehlo iba desať dní.

## 1.8 Validácia softvérových prototypov vo virtuálnej realite

Podľa knihy [14] je validácia a verifikácia jednou z najznámejších oblastí v sektore softvérového inžinierstva, ktorá existovala už od skorších dôb vývoja softvéru. Pri validácii sa zisťuje či je vyvíjaný správny produkt. Pri verifikácii je zisťované či je produkt správne vyvíjaný. Validácia a verifikácia je vykonávaná aby bolo možné sa uistiť že softvér, ktorý je vyvíjaný splňa požiadavky používateľa bez akýchkoľvek defektov. Validácia je úzko spojená so špecifikáciou požiadaviek. Požiadavky používateľov je možné validovať a práve tu je miesto, kde väčšinou vládne nejednoznačnosť a formálne metódy, použitím špecifikačných jazykov, urobili najväčší pokrok. Stále však existuje veľký rozdiel medzi tým čo používateľ chce a ako vývojár chápe to čo používateľ chce. Často tak je možné nájsť počiatočné zlyhanie systému práve pri tejto nezhode. V prípade väčšiny systémového vývoja je validácia ponechaná takmer na koniec vývoja, kedy sa používateľ stretne so systémom aby bol systém finálne schválený. To však nie je vždy prípadom, keďže existujú metódy, ktoré vyžadujú účasť používateľa pri procese vývoja.

Virtuálnu realitu je vhodné použiť na validáciu takých softvérových prototypov,



ktoré majú vykonávať nejakú činnosť autonómne. Takéto prototypy nemusia byť nutne modely strojového učenia, avšak vzhľadom na to že virtuálna realita je vhodná, ako už bolo spomenuté vyššie, na účeli ich tréovania a testovania, je možné ju používať aj na verifikáciu takýchto modelov.

## 1.9 Webová virtuálna realita

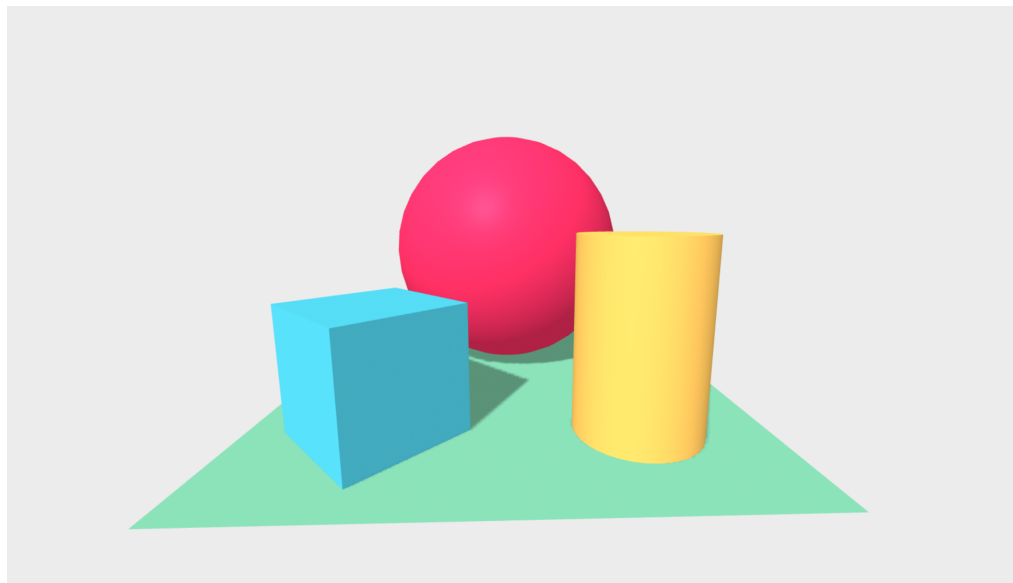
Webová virtuálna realita, ako názov naznačuje, je zážitok virtuálnej reality dostupný pre používateľov tejto technológie priamo v ich webovom prehliadači. Podľa autora knihy [15] má webová virtuálna realita hneď dve výhody v porovnaní s tradičnou virtuálnou realitou. Prvou by malo byť to, že je možné použiť technológie ako HTML5, WebGL a Javascript na vytváranie aplikácii webovej virtuálnej reality, čo by malo mať potenciálne za následok zrýchlené písanie kódu takýchto aplikácii a taktiež dostupnosť na väčšom množstve platforiem než tradičná virtuálna realita. Druhou výhodou je prístup k infraštruktúre, ktorá bola pre web už vybudovaná. Táto infraštruktúra ponúka možnosť využívať hypertextové prepojenie medzi zážitkami virtuálnej reality, mať obsah uložený priamo na internete, vytvárať aplikácie virtuálnej reality s niekoľkými používateľmi a integrovať webové údaje priamo do aplikácii virtuálnej reality.

## 1.10 Popis rámca webovej virtuálnej reality A-Frame

Kniha [16] opisuje A-Frame ako open-source rámec pre webovú virtuálnu realitu vyvíjaný MozVR tímom pod záštitou neziskovej organizácie Mozilla Foundation. Vo webovom rámci A-Frame je možné vytvárať virtuálne prostredia pomocou značkovacieho jazyka HTML. V prípade potreby je možné do A-Frame scény zasahovať aj pomocou skriptovacieho programovacieho jazyka Javascript. A-Frame podporuje širokú škálu zariadení ako napríklad inteligentné telefóny, notebooky a desktopové počítače, HTC Vive, Oculus Rift, Oculus Go, GearVR a dokonca môže byť použitý pre rozšírenú realitu. Rámec A-Frame je postavený na 3D Javascript knižnici Three.js.

Vďaka veľkému počtu používateľov a vývojárov má rámec A-Frame veľké množstvo rôznych komponentov, ktoré boli vytvorené a zverejnené touto komunitou pre zjednodušenie vývoja aplikácii webovej virtuálnej reality. Príkladom týchto komunitou vytvorených komponentov je napríklad aframe-physics-system ktorý umožňuje vo virtuálnej realite simulovať rôzne fyzikálne interakcie alebo raycas-

ter komponent ktorý ponúka možnosť na testovanie či sa lúč ktorý je vytvorený z jedného bodu a smeruje do druhého pretína s nejakým objektom. Okrem iného taktiež umožňuje získať odkaz na daný objekt a taktiež polohu bodu kde sa lúč a objekt pretli.



Obr. 1.7: Hello world, respektíve Hello WebVR v rámci A-Frame.

## 2 Syntetická časť

---

Pri vytvorení riešenia transformácie objektov reálneho sveta do virtuálneho prostredia bude použitých niekoľko rozdielnych technológií, ktoré budú spolu slúžiť na docieľenie tejto úlohy. Webová rozšírená realita bude zabezpečená využitím webového rámca A-Frame. Vo webovom rámci A-Frame bude vytvorená sada virtuálnych prostredí, na ktorej bude meraný vplyv takejto transformácie na výkon. Vzhľadom na to že webový rámec A-Frame zadarmo, open-source, má podporu pre veľké množstvo rôznych platforiem a má najväčší počet používateľov a vývojárov v porovnaní s ostatnými rámcami pre webovú virtuálnu realitu, sa zdá byť najvhodnejší pre účeli tejto práce. Na zabezpečenie transformácie reálnych objektov na virtuálne objekty bude použitý program Blender. Blender bude použitý z toho dôvodu že doplnky, ktoré k nemu boli vytvorené komunitou, umožňujú takúto transformáciu vykonávať na veľkej škále. Bude taktiež využívaná webová stránka Glitch, ktorá umožňuje hostovanie webových aplikácií. To bude využívané na to aby virtuálne scény bolo možné testovať na rôznych zariadeniach.

### 2.1 Návrh a implementácia sady virtuálnych prostredí

Pre overenie vplyvu oblasti z reálneho sveta, ktorá bola transformovaná do virtuálnej reality na výkon rôznych zariadení bolo navrhnutých niekoľko rôznych scén s rôznymi úrovňami zložitosti. Tieto scény sa líšia tým ako sú rozdelené budovy ktoré boli importované do scény, materiálmi ktoré boli aplikované na modeli a tým aký typ stromov sa v scéne nachádza. Čo sa týka materiálov sú tu tri úrovne. Prvá je jedna farba aplikovaná na všetky budovy a terén. Tak ako boli budovy importované do programu Blender doplnkom Blender-OSM. Jedinou úpravou bolo v tomto prípade predĺženie budov tak aby sa nevznášali nad terénom. Druhou úrovňou materiálov sú náhodné farby, ktoré boli aplikované na jednotlivé budovy. Na terén bola aplikovaná zelená farba tak aby pripomínala trávnatý porast. Treťou úrovňou sú materiály, ktoré mali aplikované textúry v programe Blender. Existujú taktiež tri úrovne rozdelenia modelov podľa počtu budov ktoré obsa-

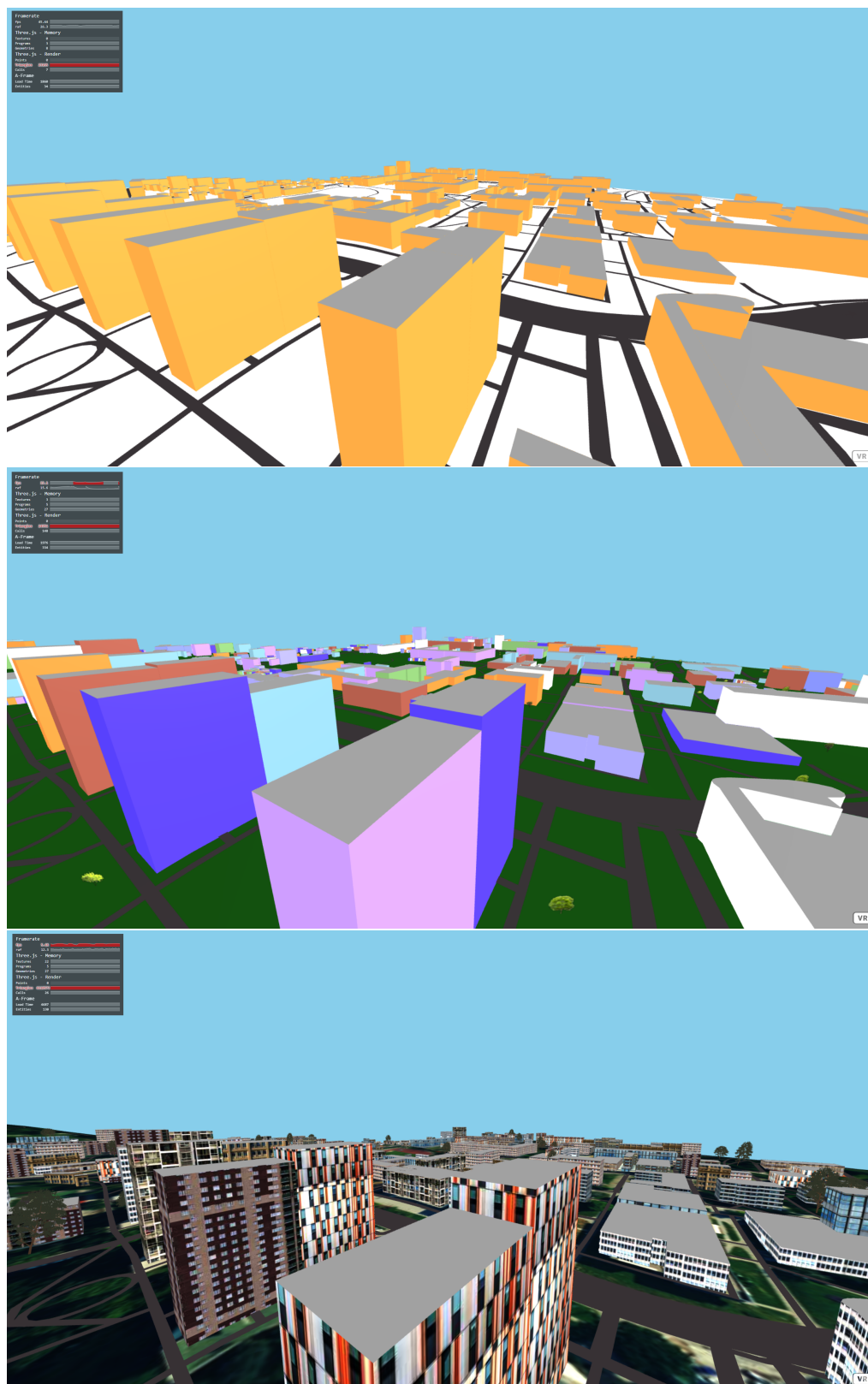
hujú. Prvou úrovňou rozdelenia sú dva modely, z ktorých obe obsahujú približne polovicu budov nachádzajúcich sa v scéne. Rozdelenie na prinajmenšom dve polovice bolo v tejto scéne nutné kvôli problémom s kolíziami. Druhou úrovňou rozdelenia budov v scéne je rozdelenie približne po dvoch budovách na jeden model. V prípade že počet budov v scéne nie je párny, môžu niektoré modely obsahovať len jednu budovu. Treťou úrovňou rozdelenia je jedna budova na jeden model. V podstate to znamená že budovy neboli zoskupené vôbec. Táto úroveň je kvôli najväčšiemu počtu volaní najnáročnejšia zo všetkých troch úrovní. Stromy ktoré sa nachádzajú v scéne majú taktiež samostatné úrovne. Prvou úrovňou je jednoducho to že stromy nie sú v scéne prítomné. Druhou úrovňou je to že na vyobrazenie stromov sú použité sprity, čo sú dvojrozmerné billboardy stromov, ktorým je nastavená rotácia podľa rotácie používateľa. Treťou úrovňou sú trojrozmerné modely stromov ktoré sú dosadené do scény. V oboch prípadoch je počet stromov dosadených do scény rovnaký. Tento počet je tristo stromov a bol zvolený z toho dôvodu, že pri nižších počtoch vyzeral stromový porast v podobe dvojrozmerných billboardov príliš riedko. Spojením týchto troch rôznych spôsobov rozdeľovania úrovní od najmenej náročných po najnáročnejšie bolo vytvorených deväť rôznych úrovní zložitosti.

Každá je meraná v troch rôznych výškach nad terénom. Prvá výška je hneď nad povrchom terénu, druhá je vo vzduchu nad terénom a tretia je vysoko vo vzduchu nad terénom. Pri týchto meraniach kamera prechádza v scéne po predvolenej dráhe a pozerá sa na stred scény.

## 2.2 Vytváranie scény vo webovej virtuálnej realite

Pre vytvorenie virtuálnych prostredí založených na reálnych prostrediach vo webovom rámci A-Frame bol použitý postup ktorý začína v 3D modelovacom programe Blender.

Program Blender bol zvolený z viacerých dôvodov. Prvým dôvodom je práve licencia programu. Blender je vydaný pod licenciou GNU General Public License alebo GPL, ktorá umožňuje používať program na akýkoľvek účel, či už komerčný, osobný alebo pri výskume a taktiež je dovolené tento program, jeho časti alebo upravené verzie voľne distribuovať. Vďaka tomu je možné voľne šíriť projekty vytvorené v tomto programe ako aj doplnky vytvorené špeciálne pre tento prog-

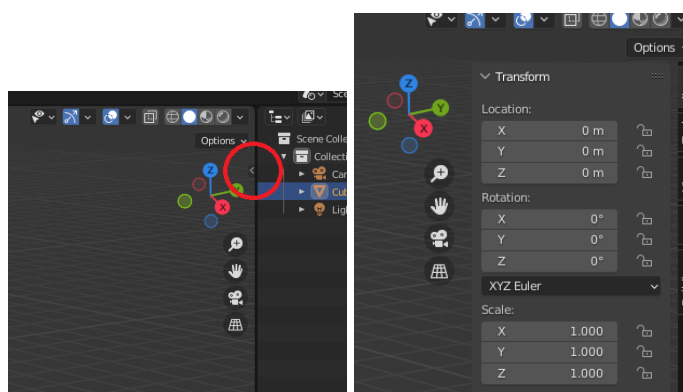


Obr. 2.1: Vzhľad jednotlivých úrovní, konkrétne úrovne 1, 4, 7.

ram. Druhým dôvodom je že Blender má veľkú komunitu používateľov a vývojárov ktorí vytvárajú doplnky do tohto programu, ktoré sú často taktiež pod

rovnakou alebo podobnou licenciou ako tento program. Niektoré z týchto doplnkov je možné používať pri transformácii objektov reálneho sveta do virtuálnych prostredí. Tretím dôvodom je to, že Blender má veľmi rozsiahlu dokumentáciu, takže aj napriek zložitosti tohto programu je stále prístupný začiatočníkom, ktorý s prácou v 3D programoch nemajú veľké množstvo skúseností. Blender je možné stiahnuť z oficiálnej stránky [www.blender.org/download/](http://www.blender.org/download/).

V programe Blender, v ktorom už bol nainštalovaný a aktivovaný doplnok Blender-OSM, je nutné stlačiť klávesu N na klávesnici alebo kliknúť na šípku ktorá sa nachádza v pravom hornom rohu plochy, v ktorej sú vykresľované 3D objekty. Po



Obr. 2.2: Šípku v pravom hornom rohu

stlačení tejto šípky alebo klávesy N sa na pravej strane plochy objaví menu s niekoľkými záložkami. V prvom rade je potrebné kliknúť na záložku Zobraziť (angl. View) a nastaviť hodnotu Koniec (angl. End), ktorá je nastavená na tisíc metrov, na vyššiu hodnotu ako napríklad stotisíc metrov. Táto hodnota určuje aká je maximálna diaľka od kamery, na ktorej budú 3D objekty vykresľované. Vzhľadom na to že je vytváraná rozsiahla scéna, je nutné aby táto hodnota bola čo najvyššia tak aby bolo vidno scénu, na ktorej sa pracuje. Následne je nutné kliknúť na záložku s názvom osm. Táto záložka predstavuje doplnok Blender-OSM, ktorý slúži na importovanie mapových údajov z reálneho sveta. Tu je najprv nutné kliknúť na tlačidlo select. Kliknutie na toto tlačidlo otvorí webovú stránku s mapou sveta. Na tejto mape je možné navigovať na ľubovoľné miesto na povrchu zeme. Po tom čo bola zvolená oblasť, ktorej údaje majú byť stiahnuté doplnkom Blender-OSM, je nutné kliknúť na tlačidlo show selection rectangle, ktoré sa nachádza v menu na ľavej strane okna prehliadača. Kliknutie na toto tlačidlo zobrazí obdĺžnik výberu oblasti na mape. Tento obdĺžnik je možné podľa potrieb manuálne zväčšiť alebo zmenšiť, čo umožňuje upresniť oblasť, ktorej údaje budú stiahnuté. V menu na ľavej strane sa taktiež nachádzajú údaje o veľkosti oblasti, ktorej údaje budú

stiahnuté, geografické koordináty tejto oblasti a krátke inštrukcie na používanie tejto stránky v anglickom jazyku. Potom čo sú údaje oblasti, ktorej údaje majú byť stiahnuté, upresnené je nutné kliknúť na tlačidlo copy, ktoré skopíruje geografické koordináty oblasti. Po tom čo sú koordináty úspešne skopírované je nutné sa vrátiť do programu Blender a v záložke doplnku Blender-OSM kliknúť na tlačidlo paste, ktoré sa nachádza v hornej časti tejto záložky. Toto tlačidlo skopíruje koordináty do doplnku Blender-OSM, ktorý ich bude následne používať na vygenerovanie scény.



Obr. 2.3: Webová stránka s mapou sveta.

Po tom čo sú koordináty zadane, je nutné vybrať prvý objekt ktorý bude importovaný. V tomto prípade to bude objekt terénu. Na tento účel slúži rozotváracia ponuka ktorá sa nachádza pod zadanými koordinátami na ľavo od tlačidla import. Tu je potrebné vybrať možnosť terrain. Následne je nutné stlačiť tlačidlo import. Po stlačení tohto tlačidla sa na ploche pre zobrazovanie 3D objektov objaví terén oblasti ktorá bola vybraná. Je taktiež možné stiahnuť textúru pre tento terén. Tá je podľa výberu buď satelitná fotografia vybranej oblasti alebo mapa z OpenStreetMap. Textúru je možné stiahnuť kliknutím na rozotváraciu ponuku a vybraním možnosti image overlay. Po tom čo bola táto možnosť zvolená je nutné vybrať aký typ textúry bude použitý. Tento typ sa vyberá v ďalšej rozotváraciej ponuke s názvom Prekrytie (angl. Overlay), ktorá sa nachádza pod prvou rozotváracou ponukou. Po výbere typu textúry je nutné kliknúť na tlačidlo import. Po kliknutí na toto tlačidlo bude textúra stiahnutá a aplikovaná na povrch terénu.

Ďalším krokom je importovať budovy ktoré sa nachádzajú vo vybranej oblasti.

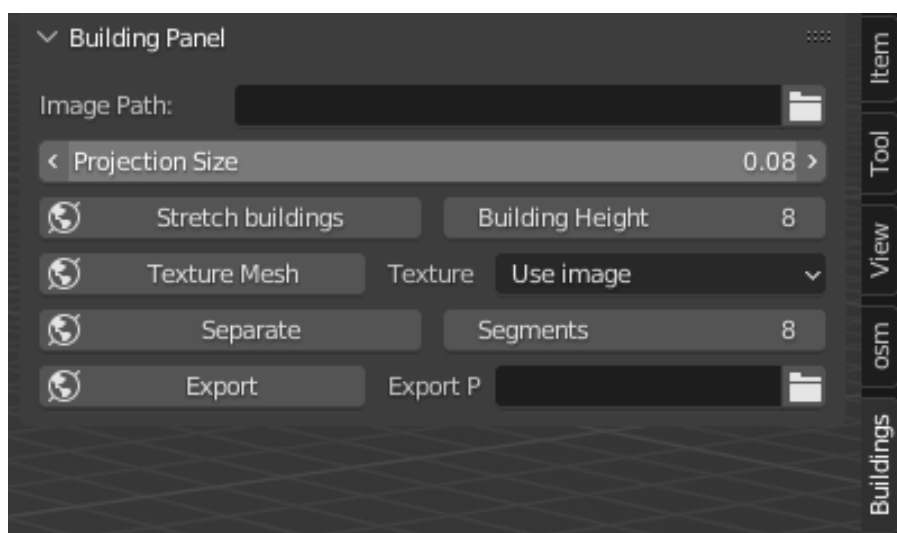
Na to je potrebné vybrať v rozotváracíj ponuke doplnku Blender-OSM možnosť OpenStreetMap. Tu je niekoľko dodatočných možností ktoré môžu byť nastavené podľa individuálnych potrieb. Možnosti ktoré sú dostupné sú napríklad objekty, ktoré budú importované. Tieto objekty sú budovy, vodné telesá, lesy, iná vegetácia, cesty, chodníky a koľajnice. Možnosť importovania budov ako trojrozmerných objektov alebo ako dvojrozmerných pôdorysov. Možnosť vybrať terén, ktorý má byť použitý na výškové dosadenie budov a či majú byť budovy importované zo servera alebo z lokálneho súboru. Možnosť vybrať tvar striech importovaných budov. Možnosť výberu výšky jednotlivých poschodí budov a či majú byť budovy importované ako jednotlivý objekt alebo každá samostatne. V prípade tejto práce boli na importovanie zvolené iba budovy, cesty, chodníky a koľajnice. Ostatné objekty neboli zvolené z dôvodu že spôsobovali artefakty, keď bola scéna pozorovaná vo webovom rámci A-Frame. Bola taktiež zvolená možnosť budovy importovať ako trojrozmerné objekty. Ako tvar striech bol zvolený plochy tvar. Možnosť importovania budov ako jedného objektu nebola vybraná vzhľadom na to, že pri úprave budov je s nimi nutné pracovať ako s jednotlivými objektmi. Výška jednotlivých poschodí bola ponechaná na predvolenú hodnotu, čo sú tri metre. Po tom čo boli potrebné možnosti nastavené, je nutné kliknúť na tlačidlo import, po stlačení ktorého budú do scény importované budovy, cesty, chodníky a koľajnice.

V prípade že cesty, chodníky a koľajnice majú byť taktiež použité v A-Frame scéne, musia byť najprv konvertované na objekt povrchovej siete. To sa vykonáva výbráním všetkých ciest, pravým kliknutím na plochu pre zobrazovanie 3D objektov, kliknutím na tlačidlo Konvertovať na (angl. Convert to) a kliknutím na tlačidlo Povrchová sieť (Mesh). Po tom čo boli cesty konvertované na povrchovú sieť je možné ich spojiť do jediného modelu. Podobne ako v predošlom kroku je nutné kliknúť na plochu pre zobrazovanie 3D objektov. Tento krát však treba zvoliť možnosť Spojiť (angl. Join).

V ďalšom kroku, ak sa to už neudialo, je nutné nainštalovať doplnok pre program Blender, ktorý bol vytvorený pre potreby tejto práce, nazvaný Buildings. V tomto doplnku je možné nastaviť niekoľko parametrov, ktoré sú potrebné pri vykonávaní rôznych úkonov na úpravu budov, ktoré za používateľa tento doplnok vykonáva. Prvým parametrom je cesta do priečinka z ktorého bude doplnok získavať textúry, ktoré budú použité pri vytváraní materiálov pre jednotlivé budovy. Druhým parametrom je veľkosť projekcie ktorá bude použitá pri premietnutí kocky.



Tretím parametrom je výška natiahnutia budov, ktorá určuje o koľko majú byť budovy zapustené do terénu aby sa nad daným terénom nevznášali. Vznášanie sa budov nad terénom bolo jedným z vizuálne najvýraznejších problémov, ktoré sa vyskytli pri prvotnom testovaní využitia mapových údajov na transformáciu reálnej scény do virtuálnej reality. [Pridať obrázok] Štvrtým parametrom je rozotváracia ponuka, v ktorej si môže používateľ vybrať aký typ materiálu má byť použitý na budovách. Na výber sú dva typy. Prvým typom je textúra ktorá používa cestu, ktorá je prvým parametrom v tomto doplnku. Druhým typom je farba, pri ktorej sú v materiáloch použité náhodne generované farby. Piatim parametrom používaním v tomto doplnku je počet segmentov, čo predstavuje na koľko jednotlivých modelov má byť scéna rozdelená respektíve do koľkých segmentov majú byť jednotlivé budovy spojené. Posledným parametrom tohto doplnku je cesta do priečinka do ktorého majú byť exportované modely uložené. Pri práci na scénach ktoré boli exportované na využitie vo webovom rámci A-Frame bolo nutné opakovane nastavovať iba hodnoty parametrov pre počet segmentov a materiály ktoré majú byť použité. Okrem toho bolo potrebné raz nastaviť cestu pre textúry a cestu pre exportovanie. V ostatných parametroch bolo možné ponechať predvolené hodnoty.



Obr. 2.4: Doplnok vytvorený pre využitie v tejto práci.

Pre exportované scény taktiež bola manuálne vytvorená mapa lesov, parkov a podobných porastov, ktorá je využívaná v scénach webového rámca A-Frame na dosadenie stromov na miesta kam patria tak, aby neprečnievali cez budovy a cesty. Na tento účel by mohol byť použitý objekt lesov a inej vegetácie, ktorý je možné

importovať pomocou doplnku Blender-OSM, avšak ten sa zdá byť v tomto prípade nepresný.

## 2.3 Návrh a implementácia skriptov pre generovanie A-Frame scény

Keďže 3D modelovací nástroj Blender ponúka možnosť používania programovacieho jazyka Python 3 na vytváranie skriptov, je možné vytvárať skripty, ktoré automatizujú opakujúce sa úkony. Skripty napísané v tomto programovacom jazyku umožňujú všestranne rozširovať funkcionality tohto softvéru. Väčšina oblastí v programe Blender, ako napríklad animácie, vykresľovanie, importovanie a exportovanie, vytváranie alebo manipulácia objektov, môžu byť ovplyvnená skriptmi.

Pri vytváraní, spracovaní a exportovaní scén v programe Blender, ktoré by následne boli použiteľné vo webovom rámci A-Frame bolo nutné vykonať veľké množstvo krokov ktoré sa často opakovali v istom poradí. Preto bolo možné veľké množstvo z týchto postupov automatizovať. Zo skriptu napísaného na automatizáciu týchto úkonov bol následne vytvorený prídavok pre program Blender, jednoducho pomenovaný Buildings (anglické slovo pre budovy).

Prvý skript ktorý bol vytvorený má za úlohu predĺžiť budovy smerom nadol tak aby boli ich najnižšie body skryté pod terénom scény. To bolo dosiahnuté tak že skript vyberie všetky objekty ktoré patria do kolekcie s názvom budovy alebo "buildings". Prepne sa z režimu objektu do režimu editácie a následne si zvolí materiál ktorý sa nachádza na indexe číslo jeden, čiže na druhej pozícii. Spolu s ním sú zvolené aj všetky strany na ktorých sa zvolený materiál nachádza. Tento materiál väčšinou predstavuje strechu budovy. V prípade že to tak nie je, je nutné túto vadu opraviť manuálne. Po tom čo je materiál zvolený, je strana na ktorej sa nachádza posunutá o určitú dĺžku smerom nahor. To efektívne vytiahne alebo predĺži danú budovu nahor. Ďalším krokom ktorý skript vykoná je zvolenie celého objektu, nie len jednej strany a posunutie celého objektu nadol o rovnakú dĺžku o ktorú bola v predošlom kroku budova predĺžená. Týmto spôsobom sú budovy efektívne predĺžené smerom nadol a ak bola zvolená správna dĺžka predĺženia, budovy by sa už nemali vznášať nad terénom. Poslednými dvoma

krokmi ktoré tento skript vykoná sú prepnutie sa z režimu editácie do režimu objektu a zrušenie výberu všetkých objektov. Tento posledný krok je nutné vykonať aby sa program vrátil do pôvodného stavu a zabráni to skriptom vo vzájomnom narušovaní.

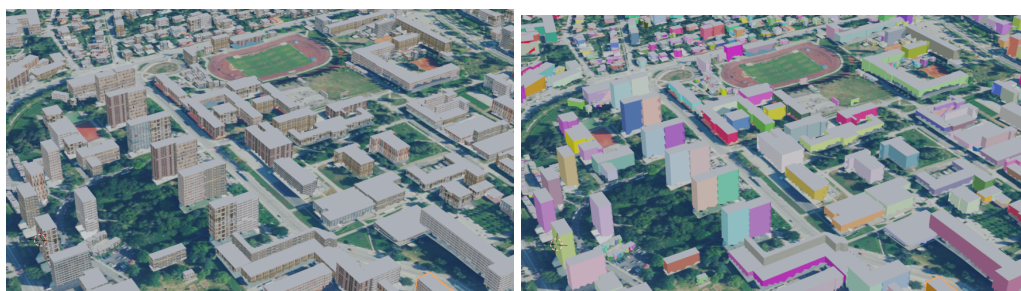


Obr. 2.5: Budovy pred a po predĺžení.

Ďalším skriptom ktorý bolo potrebné vytvoriť bol skript, ktorý slúži na vykonanie premietnutia kocky. Premietnutie kocky zmapuje povrchovú sieť na plôšky kocky, ktorá sa potom rozvinie. Pri tomto skripte je niekoľko prvých krokov rovnakých ako pri predošlom skripte na predĺženie budov nadol. Prvý odlišný krok je ten že materiál ktorý je vyberaný sa nachádza na prvej pozícii. Tento materiál predstavuje strany respektíve steny budovy. Po tom čo sú zvolené správne strany, je na nich vykonané premietnutie kocky a mierka povrchovej siete je zmenšená alebo zväčšená o vybranú hodnotu. Tento krok je nutné vykonať aby textúry ktoré neskôr budú priradené na budovách mali správne rozmery. Posledné dva kroky sú rovnaké ako v predošlom skripte. Z tohto dôvodu boli skripty predĺženia budov a premietnutia ich strán spojené do jedného a je možné ich vykonať jedným tlačidlom.

Vzhľadom na to že budovy vygenerované doplnkom Blender-OSM pomocou údajov získaných z OpenStreetMap nemajú žiadne textúry a obsahujú len jednoduché materiály s jednou farbou, bolo nutné vytvoriť skript ktorý by dokázal automatizovať proces modifikácie vzhľadu veľkého počtu budov v krátkom čase. Za týmto účelom boli vytvorené dve skripty ktoré fungujú na podobnom a zároveň veľmi jednoduchom princípe. Pri prvom je v priečinku, ktorý je zadaný, spočítaný počet textúr. Vďaka tomu skript vie koľko má vytvoriť materiálov ktoré budú dané textúry používať. Následne je vytvorený materiál. Tomuto materiálu je následne priradená textúra zo zvoleného priečinka. Tento proces je vykonaný

pre každú textúru nachádzajúcu sa v prierečniku. Textúry použité pri tomto procese musia byť vo formáte JPEG, ktorý bol zvolený kvôli jeho vysokej kompresii. Tá je nutná vzhľadom na to že v prípade, že sú budovy exportované samostatne, bude každá budova obsahovať túto textúru. Preto je vhodné aby textúry zaberali čo najmenej priestoru. Po tom čo sú materiály s textúrami vytvorené, je každej budove náhodne priradený jeden z týchto materiálov. Pri druhom je vytvorený istý počet materiálov ktorým je následne priradená náhodná farba. Počet materiálov ktoré majú byť vytvorené s náhodnou farbou je desať, avšak toto číslo môže byť ľubovoľné. Podobne ako pri predošlom skripte, materiály sú náhodne priradené každej budove.



Obr. 2.6: Budovy s textúrami a farbami.

Vzhľadom na to že na prvý pohľad mal počet budov, ktoré boli samostatne exportované a použité vo webovom rámci A-Frame dopad na výkon scény, v ktorej boli použité, bolo potrebné vytvoriť skript ktorý automaticky spojí budovy do vybraného počtu celkov. To je nutné aby bolo možné testovať dopad počtu jednotlivých 3D modelov na výkon scény. Pri tomto úkone skript pracuje so zoznamom budov ako by to bolo pole prvkov. Toto pole je rozdelené na vybraný počet polí a následne je na každom samostatnom poli vykonaná operácia spojenia. Následne je zrušený výber modelu aby sa program vrátil do pôvodného stavu.

Posledný úkon ktorý bolo potrebné automatizovať pomocou skriptov je exportovanie 3D modelov vygenerovaných doplnkom Blender-OSM a upravených skriptmi opísanými vyššie. V tomto skripte je taktiež automatizované generovanie Javascript kódu používaného na importovanie modelov budov do A-Frame rámca. Prvým krokom, ktorý je vykonaný v tomto skripte, je otvorenie alebo vytvorenie textového súboru do ktorého bude uložený vygenerovaný kód. V tomto súbore bude uložený kód, ktorý predstavuje premennú s počtom modelov ktoré sú exportované a ktoré majú byť načítané do scény. Druhým krokom je vybratie modelu z kolekcie "buildings" a jeho exportovanie vo formáte gltf. Modelu je priradené jed-

noduché meno ktoré pozostáva len z čísla ktoré je dané poradím pri exportovaní. Následne zrušený výber modelu z kolekcie tak aby nebol ten istý model exportovaný viackrát. Po tom čo je dokončené exportovanie všetkých modelov budov je do súboru zapísaný kód s premennou, ktorej je priradené číslo počtu exportovaných budov. Posledným krokom je zatvorenie tohoto súboru.

## 2.4 Kolízie vo webovom rámci A-Frame

Kolízie sú pri snahe dosiahnuť realistickosť a vierohodnosť vo virtuálnych prostrediach, ktoré sú interaktívne a v ktorých môže používateľ alebo softvér vykonávať nejakú činnosť, ktorá bude mať za následok interakciu s prostredím, nevyhnutné. Toto nie je prípad v prostrediach ktoré nie sú interaktívne, keďže jediný úkon ktorý môže používateľ vykonávať v takýchto prostrediach je pozorovanie. Vzhľadom na to že v tejto práci je potrebné preskúmať možnosť tvorby vierohodného prostredia, respektíve prostredia ktoré je transformované spolu s objektmi nachádzajúcimi sa v ňom z reálneho sveta do toho virtuálneho, je nutné používať kolízie vo vytvorenej scéne, aby používateľ nemohol prechádzať cez steny alebo terén po ktorom sa má za úlohu pohybovať.

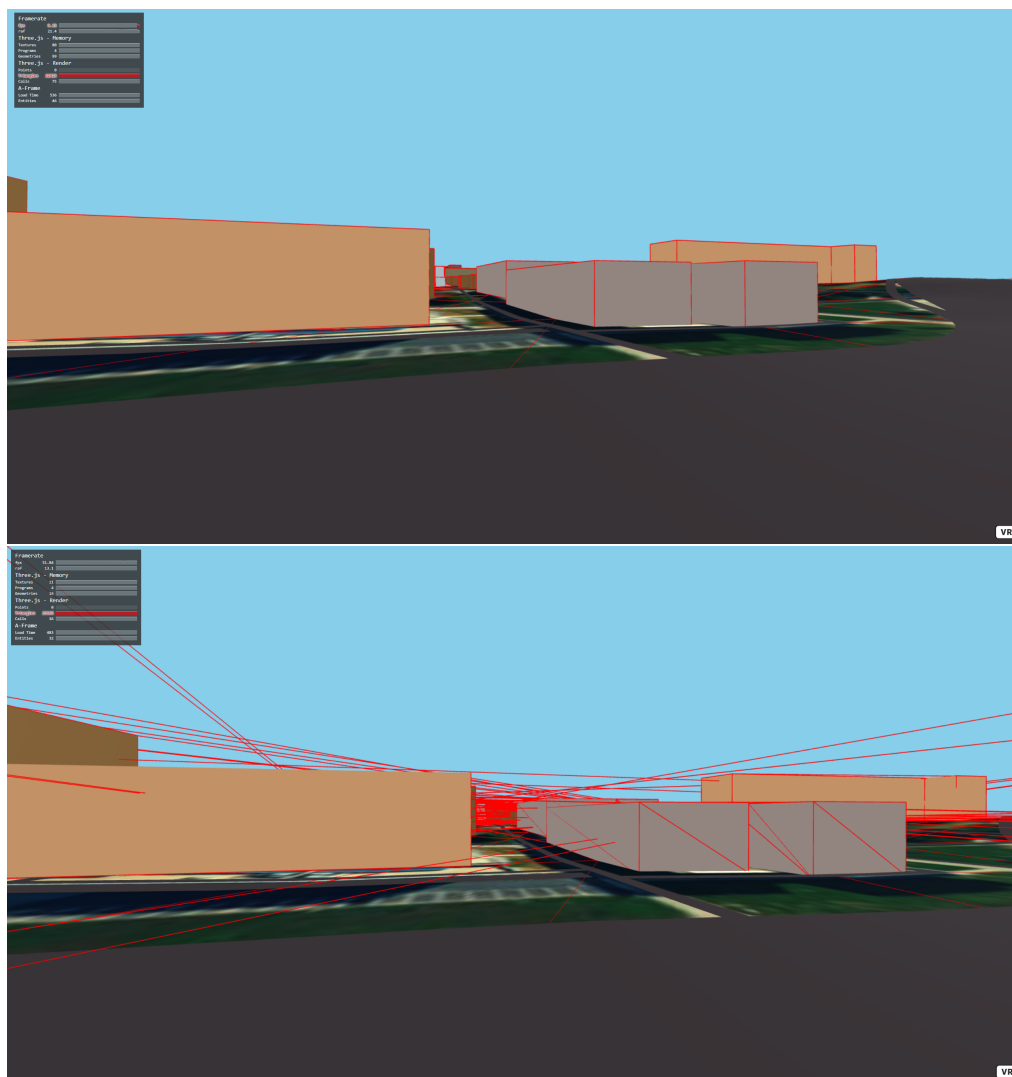
Rámec webovej virtuálnej reality A-Frame ponúka možnosť vytvárať kolízie pomocou používateľmi vyvíjanej zbierky komponentov s názvom `aframe-physics-system`, ktorá zabezpečuje integráciu fyziky pomocou `Ammo.js`. `Aframe-physics-system` komponenty ponúkajú niekoľko vymožeností ako gravitácia, trenie, kolízie s objektmi a podobne. Existuje niekoľko tvarov kolízie ktoré `aframe-physics-system` komponenty dokážu vytvoriť a ktoré je možné nastaviť v html kóde objektu na ktorý chceme kolíziu aplikovať. Tieto tvary sú aplikované ako vlastnosti (slovenský preklad anglického slova *properties*), pre komponenty statického telesa (angl. *static-body*) alebo dynamického telesa (angl. *dynamic-body*). Prvým takýmto tvarom alebo vlastnosťou je *Žiadny* (angl. *None*), ktorý ako názov napovedá, nepridá objektu žiadnu kolíziu. Druhou vlastnosťou ktorú je možné nastaviť je *Auto*, čo automaticky vyberie tvar kolízie z dostupných tvarov. Tretím tvarom je *Kocka* (angl. *Box*). Tento tvar má nízky dopad na výkon scény, hlavne v porovnaní s zložitejšími tvarmi. *Cylinder*, pridá objektu kolíziu v tvare cylindra. *Gulä* (angl. *Sphere*) pridá možnosť mať ako tvar kolízie guľu. *Trup* (angl. *Hull*) vytvorí tvar kolízie, ktorý vyzerá ako by bol model zabalený v potravinárskej fólii.

Konvexné tvary môžu mať lepší výkon než tvar siete, ale aplikácia tohoto tvaru na dynamické objekty bude mať nejaký dopad na výkon. Tvary primitívov (angl. Primitives) sú automaticky použité s korešpondujúcimi primitívnymi tvarmi objektov v rámci A-Frame. Medzi tieto primitíva patrí plocha, kocka, cylinder, guľa a podobne. Sieť (angl. Mesh) je tvar ktorý sa prispôsobí tvaru objektu na ktorý je aplikovaný. Tento tvar má minimálnu podporu, môže prechádzať cez niektoré iné tvary a má veľký dopad na výkon.

Pri testovaní boli tieto tvary vyskúšané avšak jediný tvar ktorý bol použiteľný pri zložitejšej scéne bola Sieť. V prvom rade bolo potrebné aplikovať kolíziu na terén ktorý bol exportovaný z programu Blender pomocou doplnku Blender-OSM. V tomto prípade bolo nutné použiť tvar siete, keďže terén môže mať rôzne kopce a údolia, ktoré by ostatné tvary nedokázali s presnosťou pokryť tak, aby sa po nich dalo prechádzať. Pri používaní kolízie na objekty budov bolo taktiež nutné používať tvar siete keďže iné tvary by neboli použiteľné v prípade že budovy sú exportované ako jeden model. V prípade že budovy sú importované do scény ako samostatné objekty, by bolo možné používať aj iné tvary, ako napríklad tvar kocky alebo tvar trupu. To by však vyžadovalo zmenu kódu pre importovanie modelov budov pre jednotlivé scény. Navyše pre tvar kocky by bolo nutné prechádzať každý model a manuálne nastavovať ich rotácie, čo v prípade scény s veľkým rozsahom by mohlo byť aj niekoľko stoviek modelov. Z tohto dôvodu bol zvolený tvar siete aj napriek jeho nevýhodám ktoré boli popísané vyššie.

### 2.4.1 Problémy s kolíziami

Pri aplikovaní kolízií na objekty v scéne bolo zistených niekoľko závad ktoré narušovali realistickú interakciu medzi používateľom a scénou virtuálnej reality vo webovom rámci A-Frame. Prvá záhada bola zistená pri testovaní aplikovania kolízie na budovy v scéne, ktoré boli exportované z programu Blender a importované do rámca A-Frame ako jeden model. Tu sa zdá že tvar kolízie mal problém s počtom bodov alebo s počtom hrán ktoré sa nachádzali v tomto jednotnom 3D modeli. Problém spočíval v tom že zatiaľ čo kolízia v tvare siete by mala presne kopírovať tvar objektu na ktorý je aplikovaná, kolízia pri tomto modeli, kde boli všetky budovy v scéne spojené do jedného, nekopírovala tvar budov ale spájala pravdepodobne náhodné body na povrchu modelu. Na to že je to problém s počtom bodov sa prišlo tak že bolo otestovaných niekoľko rozdielnych teórií o prí-



Obr. 2.7: Budovy zo správnymi a nesprávnymi kolíziami.

čine tohto problému. Najprv bola otestovaná možnosť vadného 3D modelu tak že bolo vyskúšaných niekoľko iných spôsobov exportovania ako napríklad iné časti mesta alebo iné formáty, do ktorých bol exportovaný model uložený. Tu sa ukázalo že tieto elementy nemali vplyv na túto vadu. Potom bola otestovaná možnosť že na vadu vplýva geografická veľkosť scény respektíve vzdialenosť medzi jednotlivými časťami modelu. Táto teória sa taktiež ukázala ako nepravdivá, keďže scéna ktorá bola vyskúšaná mala rovnaké geografické rozmery ako tá predošlá avšak menšiu hustotu budov. Po tom čo sa ukázalo že geografické rozmery nemajú vplyv na tento problém bolo otestované rozdelenie tohto jednotného modelu na dve polovice. Potom čo tieto polovice boli samostatne importované do webového rámca A-Frame tento problém zmizol. Druhým problémom ktorý bol zistený bolo to, že ak budovy, ktoré boli exportované z programu Blender a neboli exportované ako samostatné modely, dochádzalo k problému s chýbajúcimi kolí-

ziami. Kolízie ktoré chýbali boli vždy časti ktoré sa nachádzali na vrchnej strane budov. V podstate chýbala kolízia ich striech. Tu bolo testovaných niekoľko scén s rôznym počtom budov exportovaných v jednom modeli. Tu sa ukázala tendencia scén s väčším počtom budov na jeden model mať väčšie množstvo chýbajúcich kolízií než tých s menším počtom budov exportovaných v jednom modeli. Tak tiež sa ukázalo že v modeloch s menším počtom budov na jeden model, napríklad ak boli exportované dva budovy v jednom modeli, chyby v kolíziách vznikali na budovách ktoré mali komplexnejší tvar strechy. V prípade že budovy boli importované do rámca A-Frame ako samostatné modely tento problém zanikol. Tretím problémom ktorý vznikol bolo prechádzanie používateľa cez kolízie, ktoré bolo závislé od rýchlosti pohybu používateľa po scéne. Tu sa ukázalo že rýchlosť pohybu bola priamo úmerná s výskytom tohoto problému. Zdá sa že ak má používateľ nastavenú rýchlosť ktorá je väčšia ako je základná rýchlosť, čiže ak je vlastnosť komponentu ovládača pohybu s názvom rýchlosť (ang. speed) nastavená na hodnotu väčšiu ako jedna. Pri základnej rýchlosti táto záhada v ignorovaní kolízií zatiaľ nebola zaznamenaná. Avšak tento problém sa stále môže vyskytnúť ak má scéna počet snímkov, ktorý je nižší než tridsať snímkov za sekundu.

## 2.5 Importovanie budov do scény

Vzhľadom na to že je potrebná schopnosť vytvárať veľké množstvo scén s rôznymi počtami budov ktoré sa v týchto scénach nachádzajú, bol navrhnutý spôsob ako tento proces automatizovať. Bol napísaný komponent s názvom spawn-buildings ktorý je v html kóde aplikovaný na prázdny element, ktorý sa nachádza v scéne. Tento prázdny element má jedinečný identifikátor s hodnotou „buildingHolder“. Importovanie modelov budov je vykonávané nasledovne. Najprv je získaný odkaz na scénu a sú vytvorené dva prázdne elementy. Jeden element slúži ako objekt na importovanie modelu budovy, ktorá bude entitou v scéne. Druhý prázdny element slúži ako entita ktorá sa bude nachádzať priamo v scéne a bude mať model objektu ktorý bol importovaný pomocou predošlého elementu. Následne sú obom elementom priradené jedinečné identifikátory a element importovaného objektu je pridaný do scény. Potom čo boli jedinečné identifikátory pridelené, je entite pridelený importovaný objekt ako gltf model. Ďalej je nastavené či má mať entita kolízie a aký má byť ich tvar. Posledným krokom je pridanie entity, ktorá bude predstavovať budovu do scény.



### 2.5.1 Problémy s výkonom pri vyššom počte budov

Tento problém vznikol v prípade ak boli budovy importované do scény ako samostatné modely. V normálnom prípade mala scéna šesťdesiat snímkov z sekundu ale v prípade že boli budovy importované samostatne, počet snímkov klesol pod štyridsať snímkov z sekundu. Prvou teóriou o príčine tohto problému bola tá že počet objektov, ktorý bol v tomto prípade cez sedemsto, mal vplyv na počet snímkov za sekundu v scéne. Táto teória sa však nepotvrdila keďže aj pri rovnakom počte objektov na scéne, v prípade že budovy neboli v programe Blender spracované ale rovno exportované tak ako boli vygenerované doplnkom Blender-OSM, nebol zaznamenaný dopad na výkon scény. Následne bolo vyskúšané to, že budovám boli priradené pred exportovaním z programu Blender len jednoduché farby. V tomto prípade taktiež nebol problém s výkonom. Po konzultácii dokumentácie webového rámca A-Frame bolo vyskúšané použitie optimalizovaných textúr. Textúry boli zmenšené na 256 x 256 pixlov a ich veľkosť na disku bola kompresiou zmenšená na menej než pätnásť kilobajtov. Tu sa ukázalo že práve textúry, ktoré neboli optimalizované, mali vplyv na výkon scény.

## 2.6 Pohyb používateľa po scéne

Pohyb používateľa bol navrhnutý tak aby mal daný používateľ možnosť sa v scéne pohybovať po teréne, tak ako by to bolo v reálnom svete ale mal zároveň aj možnosť vznášať sa nad terénom. Možnosť vznášať sa nad terénom by mohla byť užitočná napríklad v prípadoch kde chce používateľ vidieť celú scénu naraz alebo dohliadnuť na niečo čo sa v scéne odohráva, ako na príklad pri validácii softvérových prototypov.

Pohyb po teréne bol vyriešený tak že v html kóde scény vybraný povrch ktorý má slúžiť ako zem. V prípade scén vytvorených v tejto práci to je terén, ktorý bol exportovaný z programu Blender a importovaný do scén v rámci A-Frame. Na tento terén neaplikovaný komponent s názvom collider-check. Tento komponent najprv skontroluje či má raycast, ktorý pretína objekt na ktorý komponent je aplikovaný, správny atribút id. Id je jedinečný identifikátor. V prípade že atribút je správny je uložená hodnota ktorá predstavuje bod na ktorom sa raycast a objekt na ktorom sa nachádza komponent pretli. Následne je skontrolované či miesto na ktorom sa pretli existuje. Ak toto miesto existuje je získaná pozícia používateľa, ktorá je následne zmenená tak aby sa nachádzala tesne nad povrchom terénu res-

pektíve miesta na ktorom sa nachádza bod kde raycast pretína terén.

Vznášanie bolo vyriešené tak že je raycast ktorý sa nachádza na entite používateľa odstránený. To dovoľí používateľovi voľne sa pohybovať po scéne. Prepínanie medzi týmito dvoma stavmi sa vykonáva pomocou tlačidla, ktoré bolo v tomto prípade nastavené na tlačidlo medzerníku na klávesnici.

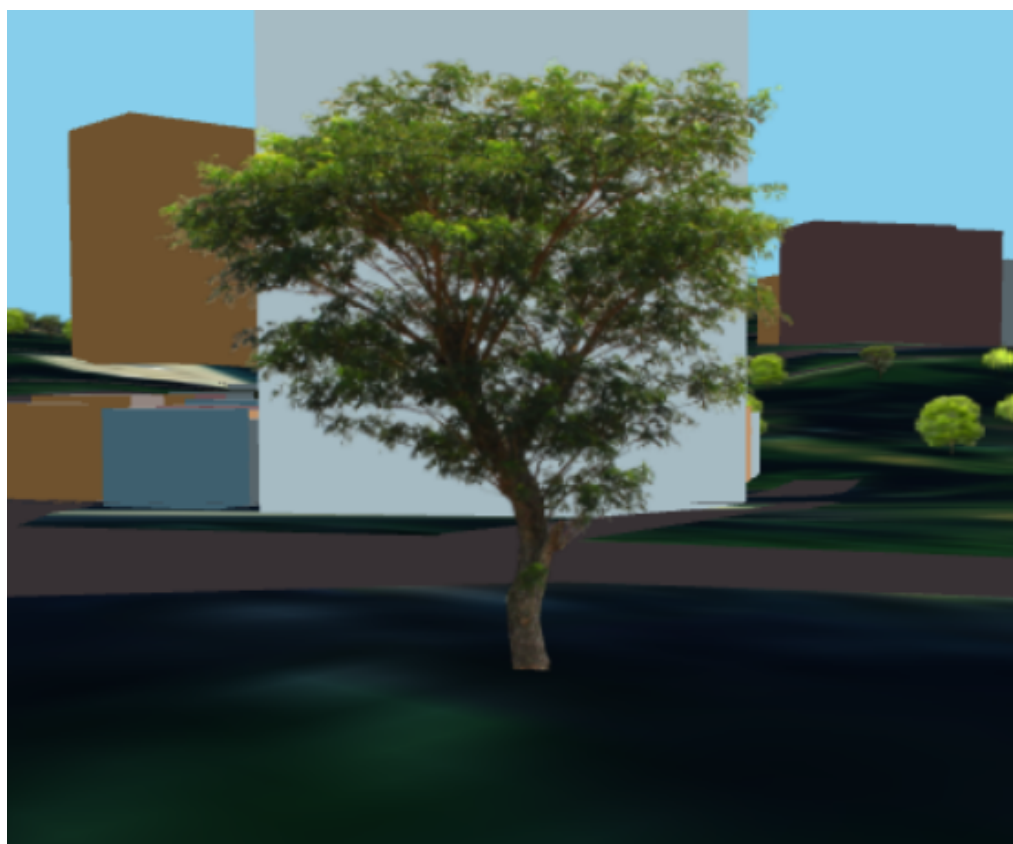
## 2.7 Stromy v A-Frame scéne

Na dosadenie stromov do scény je použitých niekoľko samostatných častí ktoré spolupracujú pri vytváraní stromového porastu v scéne. Prvou časťou je mapa stromového porastu. Táto mapa určuje kde na povrchu terénu, ktorý sa nachádza v scéne, môžu byť dosadené stromy. Pôvodne bolo zamýšľané použiť objekty lesov a inej vegetácie, ktoré je možné v programe Blender stiahnuť pomocou doplnku Blender-OSM. Tieto objekty však neboli presné a pokrývali plochu terénu, ktorá bola príliš malého rozsahu aby v scéne urobila viditeľný rozdiel vo vzhľade. Z toho dôvodu bola vytvorená v programe Blender mapa stromového porastu manuálne. Táto mapa je exportovaná z programu Blender ako model formátu gltf a importovaná do scény kde je jej viditeľnosť vypnutá. Táto mapa nemusí byť viditeľná vzhľadom na to že s ňou musí byť v mŕ interakciu len lúč raycastera. Druhou časťou je kód ktorý je využívaný na dosadenie stromového porastu na povrch scény. Tento kód najprv vytvorí neviditeľný objekt v scéne, ktorý obsahuje lúč raycastera. Tento lúč smeruje priamo nadol a vertikálna pozícia neviditeľného objektu ktorý ho obsahuje je vysoko nad terénom. Následne je pozícia x a y tohto objektu náhodne zmenená nad terénom a ak lúč príde do kontaktu s mapou stromového porastu, je na mieste, kde došlo k tomuto kontaktu dosadený strom a pozícia je znova náhodne zmenená. V prípade že lúč po zmene x a y pozície nepríde do kontaktu s mapou stromového porastu, nebude strom dosadený a pozícia bude znova náhodne zmenená. Po tom čo je dosadených tristo stromov, sa tento kód prestane opakovať a objekt obsahujúci lúč bude odstránený zo scény. Tento kód je rovnaký pre dvojrozmerné bilbordy ako aj trojrozmerné modely stromov. Rozdiel je len v tom aký model stromu je dosadzovaný do scény. Pri dvojrozmerných bilbordocho je to plocha ktorá ako textúry obsahuje jeden z troch obrázkov, z ktorých je jeden náhodne zvolený. Tejto ploche je taktiež pridaný komponent rotate-sprite ktorý jej nastavuje rotáciu. Pri trojrozmerných

modeloch je do scény dosadená inštancia trojrozmerného modelu stromu. Tejto inštancii je náhodne o pár hodnôt zmenená rotácia a škála oproti pôvodnému modelu aby vznikla vizuálna rozmanitosť medzi jednotlivými inštanciami stromov.

### 2.7.1 Rotácia dvojrozmerných billboardov

Rotácia dvojrozmerných billboardov funguje na veľmi jednoduchom princípe. Komponent rotate-sprite, ktorý bol pridaný objektu plochy obsahujúcej obrázok stromu, zistí rotáciu objektu na ktorom sa v scéne nachádza kamera. Následne tento komponent zmení rotáciu plochy ktorej bol priradený na osi y. Vďaka tomu vzniká ilúzia že plocha je stále otočená smerom na používateľa.



Obr. 2.8: Dvojrozmerný strom.

### 2.7.2 Používanie inštancii 3D modelov stromov

Používanie inštancii je technika používaná v 3D počítačovej grafike na renderovanie niekoľkých kópií toho istého modelu v scéne naraz. Tato technika sa väčšinou používa na zobrazovanie stromov, tráv a podobne. Aj keď niektoré údaje kópií modelov sú rovnaké, niektoré ako na príklad farba, pozícia, rotácia, škála, animácia a podobne, môžu byť zmenené aby nevyzerali všetky kópie rovnako.

Vo webovom rámci A-Frame je možné používať inštalácie pomocou komponentu `instanced-mesh`. Najprv musí byť importovaný, model ktorý má byť použitý na vytváranie inšancií. Tento model je následne vložený do scény a je mu pridaný komponent `instanced-mesh`. Všetkým entitám, ktoré majú byť inštanciami tohto modelu, je pridaný komponent `instanced-mesh-member` a ako hodnota tohto komponentu je nastavený jedinečný identifikátor entity ktorá obsahuje komponent `instanced-mesh`. Jednotlivé inštalácie by taktiež mali mať svoj vlastný jedinečný identifikátor. Pri používaní inštancií stromov sa však vyskytol problém s materiálmi ktoré sa nachádzali na modeli stromu. Zdá sa že komponent dokáže zobrazovať iba jeden materiál na modeli ktorý je používaný ako inštancia. Tento problém bol vyriešený zjednotením materiálov, ktoré sa na modely stromu nachádzali, do jedného.



Obr. 2.9: Trojrozmerný strom.

### 3 Meranie výkonu vytvorených scén

---

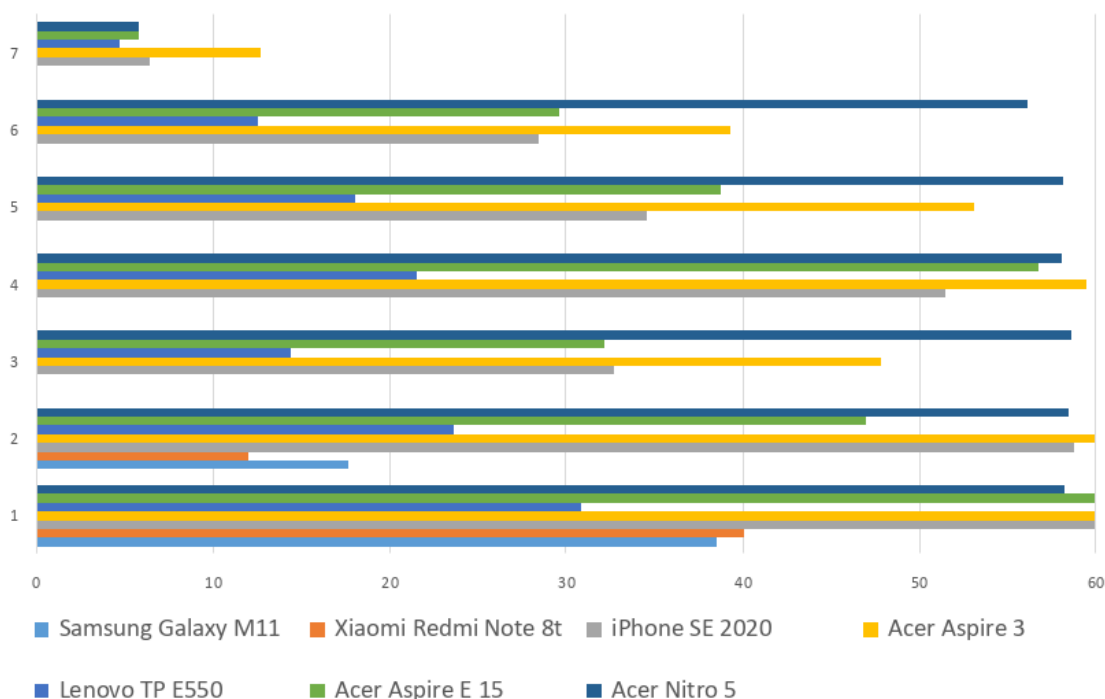
Meranie výkonu sady virtuálnych prostredí bolo vykonávané na mobilných zariadeniach ako aj na osobných počítačoch. Tieto zariadenia používali rozličný hardvér, operačné systémy ako aj webové prehliadače. Aj napriek tomu že oficiálna stránka webového rozhrania A-Frame má zoznam webových prehliadačov s ktorými funguje rozhranie A-Frame lepšie, ako napríklad Firefox a tých s ktorými funguje horšie, ktorých príkladom je webový prehliadač Safari pre iOS, cieľom webovej virtuálnej reality je dostupnosť pre širokú škálu používateľov. Z tohto dôvodu nebol braný ohľad na zariadenie, operačný systém ani prehliadač ktorý bol používaný a na všetkých zariadeniach bolo meranie výkonu scén virtuálnej reality vykonávané rovnakým spôsobom. Zariadenia na ktorých bolo vykonávané meranie súboru sú Acer Aspire 3, Lenovo TP E550, Acer Aspire E 15, Acer Nitro 5, Samsung Galaxy M11, Xiaomi Redmi Note 8t a iPhone SE 2020. Ako už bolo popísané vyššie, meranie bolo vykonávané na deviatich úrovniach zložitosti a v troch rozličných výškach nad terénom scény, s cieľom dostať na obrazovku čo najväčšie časti scény pri pohybe v danej scéne. Meranie bolo vykonávané so zapnutými kolíziami, keďže pri normálnom používaní by kolízie boli taktiež zapnuté. Meranie pre každú výšku trvalo približne desať minút. Čo znamená pol hodina bola potrebná na odmeranie jednej úrovne zložitosti. Celkovo meranie výkonu jedného zariadenia by malo zabráť štyri a pol hodiny ale vzhľadom na prehrievanie zariadení a nutnosť nechať ich z času na čas vychladnúť spojeným s tým že niektoré zariadenia nezvládli udržať použiteľný počet snímok za sekundu v istých úrovniach zložitosti zmenilo čas, ktorý bol potrebný na odmeranie dopadu zložitosti na výkon, zo zariadenia na zariadenie.

Počet snímok za sekundu bol meraný kódom, ktorý získal tento počet zo scény a uložil ho do poľa každých sto milisekúnd. Pôvodne bol plán použiť iné vlákno na meranie počtu snímok za sekundu ale zdá sa že web workers, čo je API ktoré zabezpečuje v jazyku Javascript používanie viacerých vlákien, nemá priami prístup k DOM elementom. Potom čo je v poli šesťtisíc hodnôt počtu snímok za se-

kundu, je toto pole prevedené do CSV súboru, ktorý je následne stiahnutý. Tieto súbory boli nakoniec spracované tak že z každej úrovne zložitosti boli z priemer-  
ných hodnôt snímkov za sekundu v rôznych výškach scény urobené priemery. V  
prípade že počet snímkov padol pod hodnotu pätnásť snímkov za sekundu, bolo  
meranie zastavené na danej úrovni.

Zariadenie	Úroveň 1	Úroveň 2	Úroveň 3	Úroveň 4	Úroveň 5	Úroveň 6	Úroveň 7	Úroveň 8	Úroveň 9
Samsung Galaxy M11	38,5	17,6							
Xiaomi Redmi Note 8t	40	12							
iPhone SE 2020	59,9	58,7	32,7	51,4	34,5	28,4	6,4		
Acer Aspire 3	59,9	59,9	47,8	59,4	53,1	39,3	12,7		
Lenovo TP E550	30,8	23,6	14,4	21,5	18	12,5	4,7		
Acer Aspire E 15	59,9	46,9	32,1	56,7	38,7	29,6	5,8		
Acer Nitro 5	58,2	58,4	58,6	58	58,1	56,1	5,8		

Odmerané hodnoty snímkov za sekundu



Obr. 3.1: Tabuľka a graf vyhodnotenia meraní.

Ako je vidieť na grafe a tabuľke vyššie, žiadne zo zariadení nedokázalo zvládnuť všetkých deväť úrovní zložitosti. To bolo prípadom aj pre zariadenia ktorým sa až do šiestej úrovne darilo dobre. Na prvý pohľad sa zdá že to je spôsobované príliš veľkým počtom trojrozmerných modelov, aj napriek tomu že sú to inštan-  
cie.

## 4 Záver

---

Cieľom tejto práce, ako už názov napovedá, bolo preskúmať možnosti transformácie objektov ktoré pochádzajú z reálneho sveta do virtuálnych prostredí. Prvou úlohou, ktorú bolo nutné vykonať k dosiahnutiu tohto cieľa, bolo vykonať analýzu súčasne používaných prístupov k tejto problematike. Tieto riešenia mali v najlepšom prípade byť voľne dostupné a využiteľné v rozšírenej realite. Výsledkom splnenia tejto úlohy je analytická časť tejto práce, v ktorej boli preskúmané a vyskúšané rôzne voľne dostupné riešenia. Ďalším krokom k splneniu tohto cieľa bolo navrhnúť a opísať konkrétny postup vykonávania takejto transformácie. Tento postup mal byť podložený vytvorením virtuálnych prostredí, v ktorých by bolo možné overiť aký vplyv má zložitosť transformovaných objektov na výkon v prostredí webovej rozšírenej reality. Postup vytvárania týchto virtuálnych prostredí sa podarilo navrhnúť a bol popísaný v syntetickej časti tejto práce. Virtuálne prostredia, v ktorých malo byť uskutočnené overovanie vplyvu zložitosti transformovaných objektov boli taktiež vytvorené. Posledným krokom bolo vykonanie meraní, ktoré overili aký má vplyv zložitosť transformovaných objektov na výkon zariadení na ktorých bolo meranie uskutočnené. Zatiaľ čo všetky tieto úlohy boli splnené a bol vytvorený postup, ktorý umožňuje rýchlu iteráciu transformácie objektov reálneho sveta do virtuálnych prostredí, je stále možné tento postup zlepšiť, či už zvýšením kvality objektov ktoré boli transformované do virtuálnych prostredí, lepšou optimalizáciou a zlepšením výkonu alebo rozšírením tohto postupu tak aby bol skvalitnený zážitok v scénach rozšírenej reality vytvorených týmto postupom.

# Literatúra

---

1. SEQUEIRA, V.; NG, K.; WOLFART, E.; GONÇALVES, J.G.M.; HOGG, D. Automated reconstruction of 3D models from real environments. *ISPRS Journal of Photogrammetry and Remote Sensing*. 1999, roč. 54, č. 1, s. 1–22. ISSN 0924-2716. Dostupné z DOI: [https://doi.org/10.1016/S0924-2716\(98\)00026-4](https://doi.org/10.1016/S0924-2716(98)00026-4).
2. SCHÖPS, Thomas; SATTler, Torsten; HÄNE, Christian; POLLEFEYS, Marc. Large-scale outdoor 3D reconstruction on a mobile device. *Computer Vision and Image Understanding*. 2017, roč. 157, s. 151–166. ISSN 1077-3142. Dostupné z DOI: <https://doi.org/10.1016/j.cviu.2016.09.007>. Large-Scale 3D Modeling of Urban Indoor or Outdoor Scenes from Images and Range Scans.
3. TANCIK, Matthew; CASSER, Vincent; YAN, Xincheng; PRADHAN, Sabeek; MILDENHALL, Ben; SRINIVASAN, Pratul; BARRON, Jonathan T.; KRETZSCHMAR, Henrik. Block-NeRF: Scalable Large Scene Neural View Synthesis. *arXiv*. 2022.
4. MUNKBERG, Jacob; HASSELGREN, Jon; SHEN, Tianchang; GAO, Jun; CHEN, Wenzheng; EVANS, Alex; MÜLLER, Thomas; FIDLER, Sanja. Extracting Triangular 3D Models, Materials, and Lighting From Images. *CoRR*. 2021, roč. abs/2111.12503. Dostupné z arXiv: 2111.12503.
5. KRAUS, Karl. *Photogrammetry: geometry from images and laser scans*. Zv. 1. Walter de Gruyter, 2007.
6. GRIWODZ, Carsten; GASPARINI, Simone; CALVET, Lilian; GURDJOS, Pierre; CASTAN, Fabien; MAUJEAN, Benoit; DE LILLO, Gregoire; LANTHONY, Yann. AliceVision Meshroom: An open-source 3D reconstruction pipeline. In: *Proceedings of the 12th ACM Multimedia Systems Conference*. 2021, s. 241–247.



7. GIRINDRAN, Renoy; BOYD, Doreen S; ROSSER, Julian; VIJAYAN, Dhanya; LONG, Gavin; ROBINSON, Darren. On the Reliable Generation of 3D City Models from Open Data. *Urban Science*. 2020, roč. 4, č. 4. ISSN 2413-8851. Dostupné z doi: 10.3390/urbansci4040047.
8. HAKLAY, Mordechai; WEBER, Patrick. OpenStreetMap: User-Generated Street Maps. *IEEE Pervasive Computing*. 2008, roč. 7, č. 4, s. 12–18. Dostupné z doi: 10.1109/MPRV.2008.80.
9. HASHIMOTO, Taisuke; SAITO, Masaki. Normal Estimation for Accurate 3D Mesh Reconstruction with Point Cloud Model Incorporating Spatial Structure. In: *CVPR workshops*. 2019, zv. 1.
10. SACKS, Rafael; PERLMAN, Amotz; BARAK, Ronen. Construction safety training using immersive virtual reality. *Construction Management and Economics*. 2013, roč. 31, s. 1005–1017. Dostupné z doi: 10.1080/01446193.2013.828844.
11. GREENGARD, Samuel. *Virtual reality*. Mit Press, 2019.
12. MARTINEZ, Mark; SITAWARIN, Chawin; FINCH, Kevin; MEINCKE, Lenart; YABLONSKI, Alex; KORNHAUSER, Alain L. Beyond Grand Theft Auto V for Training, Testing and Enhancing Deep Learning in Self Driving Cars. *CoRR*. 2017, roč. abs/1712.01397. Dostupné z arXiv: 1712.01397.
13. PENG, Xue Bin; GUO, Yunrong; HALPER, Lina; LEVINE, Sergey; FIDLER, Sanja. ASE: Large-scale Reusable Adversarial Skill Embeddings for Physically Simulated Characters. *ACM Trans. Graph.* 2022, roč. 41, č. 4.
14. DASSO, Aristides. *Verification, validation and testing in software engineering*. IGI Global, 2006.
15. PARISI, Tony. *Learning virtual reality: Developing immersive experiences and applications for desktop, web, and mobile*. "O'Reilly Media, Inc.", 2015.
16. NEELAKANTAM, Srushtika; PANT, Tanay. Introduction to A-Frame. In: *Learning Web-based Virtual Reality: Build and Deploy Web-based Virtual Reality Technology*. Berkeley, CA: Apress, 2017, s. 17–38. ISBN 978-1-4842-2710-7. Dostupné z doi: 10.1007/978-1-4842-2710-7\_4.

# Zoznam príloh

---

**Príloha A** CD médium – záverečná práca v elektronickej podobe,

**Príloha B** Používateľská príručka