

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

**Vylepšenie používateľského rozhrania
virtuálneho prostredia pre vizualizáciu
experimentálnych údajov**

Diplomová práca

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

**Vylepšenie používateľského rozhrania
virtuálneho prostredia pre vizualizáciu
experimentálnych údajov**

Diplomová práca

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

Košice 2025

Bc. Natanael Prokop

Abstrakt v SJ

Táto diplomová práca sa zameriava na zlepšenie používateľského zážitku (UX) v experimentálnom nástroji NDMVR (N-DiMensional Virtual Reality). Hlavným cieľom je optimalizácia používateľského rozhrania a rozšírenie možností interakcie s VR komponentom, čím sa zvýši flexibilita aj rozsah jeho využitia. Práca kladie dôraz na minimalizáciu času potrebného na testovanie UX a zabezpečenie plynulej interakcie s VR prostredím, a to nielen počas aktívneho používania, ale aj pri testovaní.

Navrhované riešenie umožňuje efektívnejšiu analýzu dát a centralizované ovládanie nástroja. Obsahuje rozhranie, ktoré podporuje testovanie UX prostredníctvom vizualizácie testovacích scenárov, sledovania času vykonávania a manuálneho zadávania spätnej väzby po ich ukončení. Týmto spôsobom sa zefektívňuje vyhodnocovanie vplyvu zmien v prostredí na používateľský zážitok. Úvodná časť práce analyzuje existujúce technológie a metódy merania UX vo VR a identifikuje kľúčové faktory ovplyvňujúce používateľský zážitok.

Na základe analýzy boli identifikované nedostatky súčasnej verzie NDMVR a navrhnuté riešenia na ich odstránenie. V rámci týchto zmien bolo vytvorené nové používateľské rozhranie, ktoré reaguje na identifikované potreby a efektívne podporuje rôzne formy použitia nástroja, vrátane UX testovania. Nové rozhranie bolo overené počas výročného podujatia CERN na Slovensku, kde výsledky potvrdili jeho efektivitu, identifikovali ďalšie možnosti rozšírenia a viedli k optimalizovanej verzii NDMVR s lepšou interakciou a vizualizáciou dát.

Práca prináša novú metodiku merania UX v NDMVR a prispieva k jeho zlepšeniu prostredníctvom optimalizácie testovacích procesov a efektívnejšieho spracovania spätnej väzby. Na základe testovania bola vytvorená nová, vylepšená verzia NDMVR, ktorá zohľadňuje získanú spätnú väzbu a ponúka efektívnejšie riešenia pre UX.

Kľúčové slová v SJ

Používateľské rozhranie (UI), Virtuálna realita (VR), Vizualizácia údajov, Používateľská skúsenosť (UX), Testovanie UX, NDMVR, Evaluácia VR aplikácií, Používateľská angažovanosť, Experimentálne dáta, A-Frame, React, CERN

Abstrakt v AJ

This thesis focuses on improving the user experience (UX) in the experimental tool NDMVR (N-Dimensional Virtual Reality). The main goal is to optimize the

user interface and expand the interaction possibilities with the VR component, thereby increasing its flexibility and range of use. The thesis emphasizes minimizing the time required for UX testing and ensuring smooth interaction with the VR environment, not only during active use but also during testing.

The proposed solution enables more efficient data analysis and centralized control of the tool. It includes an interface that supports UX testing through the visualization of test scenarios, tracking execution time, and manually entering feedback after their completion. In this way, it streamlines the evaluation of the impact of changes in the environment on the user experience. The introductory section of the thesis analyzes existing technologies and methods for measuring UX in VR and identifies key factors affecting the user experience.

Based on the analysis, the shortcomings of the current version of NDMVR were identified, and solutions were proposed to address them. As part of these changes, a new user interface was created, which responds to the identified needs and effectively supports various forms of tool usage, including UX testing. The new interface was tested during the annual CERN event in Slovakia, where the results confirmed its effectiveness, identified further expansion opportunities, and led to an optimized version of NDMVR with improved interaction and data visualization.

This thesis introduces a new methodology for measuring UX in NDMVR and contributes to its improvement through the optimization of testing processes and more efficient handling of feedback. Based on the testing, a new, improved version of NDMVR was created, which takes into account the feedback received and offers more effective solutions for UX.

Kľúčové slová v AJ

User Interface (UI), Virtual Reality (VR), Data Visualization, User Experience (UX), UX Testing, NDMVR, VR Application Evaluation, User Engagement, Experimental Data, A-Frame, React, CERN

Bibliografická citácia

PROKOP, Natanael. *Vylepšenie používateľského rozhrania virtuálneho prostredia pre vizualizáciu experimentálnych údajov*. Košice: Technická univerzita v Košiciach, Fakulta elektrotechniky a informatiky, 2025. 89s. Vedúci práce: Ing. Štefan Korečko PhD

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

ZADANIE
DIPLOMOVEJ PRÁCE

Študijný odbor: **Informatika**

Študijný program: **Informatika**

Názov práce:

Vylepšenie používateľského rozhrania virtuálneho prostredia pre vizualizáciu experimentálnych údajov

User Intefrace Improvements of Virtual Environment for Experimental Data Visualization

Študent:

Bc. Natanael Prokop

Školiteľ:

Ing. Štefan Korečko, PhD.

Školiace pracovisko:

Katedra počítačov a informatiky

Konzultant práce:

Pracovisko konzultanta:

Pokyny na vypracovanie diplomovej práce:

1. Analyzovať súčasný stav webového komponentu NDMVR pre vizualizáciu údajov z fyzikálnych experimentov vo virtuálnej realite, najmä z hľadiska konfigurovateľnosti jeho používateľského rozhrania.
2. Komponent NDMVR upraviť pre používateľské testovanie, testovanie realizovať a vyhodnotiť.
3. Na základe analýzy a testovania navrhnúť a implementovať vybrané vylepšenia komponentu NDMVR.
4. Návrh a implementáciu koordinovať s ďalšími riešiteľmi komponentu NDMVR.
5. Implementované vylepšenia vyhodnotiť a vypracovať dokumentáciu podľa pokynov vedúceho práce.

Jazyk, v ktorom sa práca vypracuje: slovenský

Termín pre odovzdanie práce: 17.04.2025

Dátum zadania diplomovej práce: 31.10.2024



n.z. P. Vokorokos

prof. Ing. Liberios Vokorokos, PhD.

dekan fakulty

Čestné vyhlásenie

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

Košice, 17.4.2025

.....

Vlastnoručný podpis

Podakovanie

Rád by som vyjadril svoju úprimnú vďaku vedúcemu mojej diplomovej práce, Ing. Štefanovi Korečkovi, PhD., za jeho odborné usmernenie, cenné rady a podporu počas celého procesu tvorby tejto práce. Veľké podakovanie patrí aj doménovému expertovi, RnDr. Martinovi Vaľovi, PhD., za jeho neoceniteľné znalosti a skúsenosti, ktoré výrazne prispeli k rozvoju mojej práce. Ich konštruktívna spätná väzba, ochota pomôcť a profesionálny prístup mi boli veľkou oporou a pomohli mi úspešne dokončiť túto prácu.

Zároveň by som rád poďakoval svojej rodine a priateľke za ich neustálu podporu a povzbudenie počas celého môjho štúdia.

Na záver nesmiem zabudnúť ani na *Donalda E. Knutha* a *Leslieho Lamportu* za vytvorenie typografického systému $\text{\LaTeX}$, ktorý mi značne uľahčil písanie tejto práce.

Obsah

Úvod	1
1 Analytická časť - technológie	4
1.1 WebGL: Základ pre 3D grafiku v prehliadači	4
1.2 Three.js: Jadro 3D vizualizácie	4
1.3 WebXR: API pre virtuálnu a rozšírenú realitu na webe	5
1.4 A-Frame: Framework pre VR a 3D web aplikácie	6
1.5 React: Základný prehľad a využitie v projekte	7
1.6 Využitie ROOTu vo vedeckej analýze	7
1.6.1 JSRoot: Webová vizualizácia ROOT dát	9
2 Analýza prístupov a aspektov merania používateľského zážitku	10
2.1 Klasifikácia UX testovania	10
2.2 Výber sledovaných oblastí UX testovania	11
2.2.1 Sledované dimenzie UX testovania a odpovede skúmania . .	11
2.2.2 Konkretizácia UX faktorov na základe skúmaných dimenzií	14
2.2.3 Faktory kvality systému	15
2.2.4 Prepojenie univerzálnych faktorov s faktormi kvality systému	16
2.2.5 Spoločná skupina meraných faktorov	18
2.3 Rady a odporúčaná forma sledovania UX vo VR systémoch	18
2.3.1 Odporúčaná forma testovania UX vo VR systémoch	18
2.3.2 UX rady a konvencie pre VR systémy a adekvátne pokryt- nutie, sledovanie UX	19
2.4 Sumatívne meranie v praxi: Efektívne hodnotenie použiteľnosti .	20
2.4.1 Dotazník System Usability Scale (SUS)	20
2.5 Sumatívne meranie v praxi: Neobvyklé prístupy k hodnoteniu angažovanosti	21
2.5.1 Refined User Engagement Scale (UES-LF)	22
2.5.2 UES Short Form (UES-SF)	22

2.5.3	Faktory angažovanosti	22
2.5.4	Výber vhodnej verzie merania angažovanosti	23
2.5.5	Spôsob hodnotenia UES	24
3	Analytická časť - NDMVR	25
3.1	Možnosti a využitie komponentu NDMVR	25
3.2	Aktuálny stav riešenia	26
3.2.1	Ovládacie prvky riešenia	27
3.3	Stav používateľského rozhrania a návrh ďalších úprav	29
4	Príprava prostredia NDMVR na UX testovanie	31
4.1	Pohyb v prostredí NDMVR	31
4.1.1	Opis nedostatkov a návrh zmien	31
4.1.2	Implementácia a vylepšenie ovládačov pre pohyb kamery	33
4.2	UI menu v prostredí VR	37
4.2.1	Opis nedostatkov poskytovaného UI a návrh zmien	37
4.2.2	Prístup k návrhu používateľského rozhrania v prostredí A-frame	38
4.2.3	Grafický dizajn a funkčné rozmiestnenie UI menu	39
4.2.4	Implementácia navrhnutého UI menu a jeho optimalizácia	40
4.2.5	Opis šablón (templates) pre UI menu v NDMVR	42
4.2.6	Znovupoužiteľnosť a rozšíriteľnosť používateľského rozhrania (UI)	45
5	Realizácia a hodnotenie testovania respondentov	47
5.1	Návrh metodiky testovania	47
5.1.1	Integrácia hodnotiacich metrík do priebehu testovania	49
5.1.2	Detailný popis krokov a metodiky testovania	49
5.2	Scenár testovania	51
5.3	Záverečné hodnotenie výsledkov	52
5.3.1	Hodnotenie celkovej angažovanosti	53
5.3.2	Hodnotenie vnímanej použiteľnosti	54
5.3.3	Hodnotenie estetickej pritažlivosti	54
5.3.4	Hodnotenie sústredenej pozornosti	55
5.3.5	Hodnotenie faktoru odmeny	56
5.3.6	Vyhodnotenie jednotlivých krokov scenára	57
5.3.7	Závery hodnotenia	60

6	Nedostatky odhalené v rámci testovania a ich odstránenie	62
6.1	Migrácia funkcionalít VR tabletu	62
6.1.1	Analýza použitých riešení tabletu NDMVR	62
6.1.2	Odhalené nedostatky aktuálneho riešenia a ich dôvody . . .	63
6.1.3	Hodnotenie aktuálneho riešenia respondentmi	64
6.1.4	Návrh odstránenia nedostatkov prostredníctvom nového riešenia	64
6.1.5	Implementácia konfigurácie tabletu v rámci UI menu zobrazovania	65
6.1.6	Implementácia konfigurácie a zobrazovania tabletu	65
6.1.7	Zvýšenie adaptability konfigurácie tabletu a celého UI Menu	67
6.2	Optimalizácia vizuálnej prezentácie scény	69
6.2.1	Návrh vizuálnej úpravy scény	69
6.2.2	Implementácia vizuálnej úpravy scény	69
6.3	Modifikácia vizualizácie histogramu	70
6.3.1	Návrh prispôsobenia dizajnu histogramu	71
6.3.2	Vizualizácia histogramu	71
7	Vyhodnotenie	74
7.1	Zhrnutie priebehu hodnotenia NDMVR	74
7.2	Retrospektívne porovnanie UX hodnotenia	75
7.3	Vyhodnotenie priebehu vykonávania úloh a správania respondentov	77
7.3.1	Časová náročnosť úloh	78
7.3.2	Pozorovania a problémy počas testovania	78
7.3.3	Subjektívne hodnotenie a spätná väzba	79
7.4	Riešenie zistených nedostatkov: Prijaté opatrenia	80
7.4.1	Nedostatočná podpora používateľského rozhrania	80
7.4.2	Vylepšenie vizuálnej konzistencie a používateľskej interakcie v NDMVR	81
7.4.3	Nedostatočná funkcionalita a vizuálne spracovanie tabletu .	82
7.4.4	Vylepšenie vizuálneho realizmu a modernizácia dizajnových prvkov	82
7.4.5	Nová scéna v prostredí	82
7.4.6	Vylepšené ovládanie pohybu	83
7.5	Zhrnutie vyhodnotenia	83
8	Záver	84

Literatúra	86
Zoznam príloh	90

Zoznam obrázkov

1.1	Príklad ROOT vizualizácie súboru údajov, prispôbeného dvojrozmernou funkciou a zobrazeného ako farebný povrch. (zdroj: [12])	8
2.1	SUS (zdroj: [23])	21
2.2	UES-SF (zdroj: [25])	23
3.1	Príklad vizualizácie histogramov typu TH3 a TH2 v NDMVR (zdroj: NDMVR)	27
3.2	Desktop zobrazenie NDMVR (zdroj: NDMVR)	28
3.3	Demo projekcia: Kolaboratívna VR (zdroj: NDMVR)	29
4.1	Grafický prehľad hierarchickej komponentovej štruktúry riešenia UI Menu primitív v A-Frame	39
4.2	Oculus UI domovské prostredie (zdroj: [31])	40
4.3	Grafický prehľad hierarchickej komponentovej štruktúry riešenia UI Menu v Reacte	42
4.4	Vytvorený template pre Music Player (zdroj: NDMVR)	43
4.5	Vytvorený template pre sledovanie úspešnosti krokov alebo úkonov (zdroj: NDMVR)	44
4.6	Vytvorený template pre Music Player (zdroj: NDMVR)	44
4.7	Vytvorený komponent pre webové vyhľadávanie, praktická ukážka spracovania klávesového vstupu od používateľa (zdroj: NDMVR)	45
5.1	Scenár testovania	51
5.2	Celkový prehľad výsledkov UES	53
5.3	Priemerná vnímaná použiteľnosť (FA-UES) získaná testovaním	54
5.4	Priemerná estetická pritažlivosť (AE-UES) získaná testovaním	55
5.5	Priemerná sústredená pozornosť (PU-UES) získaná testovaním	56
5.6	Priemerný faktor odmeny (RW-UES) získaný testovaním	57

5.7	Celkový prehľad časových intervalov jednotlivých blokov scenára testovania	58
5.8	Priemerný čas vykonávania scénara testovania	58
5.9	Priemerný čas pri riešení bloku č.1 scenára testovania	59
5.10	Priemerný čas pri riešení bloku č.2 scenára testovania	59
5.11	Priemerný čas pri riešení bloku č.3 scenára testovania	59
5.12	Priemerný čas pri riešení bloku č.4 scenára testovania	60
6.1	Viacúrovňové in-line zobrazenie tabletu NDMVR (zdroj: NDMVR)	63
6.2	Vytvorený template pre typ výber (select), konkrétne výber scény histogramu (zdroj: NDMVR)	66
6.3	Ukážka templatu pre typ rozpätia (range), konkrétne výšky binu (zdroj: NDMVR)	67
6.4	Ukážka nastavitel'ného umiestnenia okna UI menu pre potreby analýzy, spolu s možnosťou listovania výberu scény. (zdroj: NDMVR)	68
6.5	Tmavé zobrazenie s podporou bodového svetla (zdroj: NDMVR) .	70
6.6	Aktualizovaný vizuál histogramu v svetlom režime scény (zdroj: NDMVR)	71
6.7	NDMVR scéna: Nastavenie tmavšieho osvetlenia s úpravou binov a osí pre svetelné orámovanie(zdroj: NDMVR)	72
6.8	Zobrazenie UI menu pri minimálnom osvetlení novej scény (zdroj: NDMVR)	72
7.1	Priemerná angažovanosť vo verziách NDMVR podľa UX testovania (UES-SF)	75
7.2	Priemerná estetická príťažlivosť spolu s faktorom odmeny pri používaní rôznych verzií NDMVR v závislosti od UX testovania (UES-AE, UES-RW)	76
7.3	Priemerná vnímaná použiteľnosť spolu so sústredenou pozornosťou pri používaní rôznych verzií NDMVR v závislosti od UX testovania (UES-PU, UES-FA)	77
7.4	Krivka osvojovania ovládania a orientácie v NDMVR	78

Zoznam tabuliek

Úvod

V posledných rokoch sa virtuálna realita (VR) stala kľúčovým nástrojom v rôznych oblastiach vedy a výskumu. Špeciálne v oblasti fyziky, a to najmä v jadrovom výskume, kde objem získaných dát často presahuje schopnosti tradičných analytických prístupov. V tejto súvislosti sa virtuálna realita stáva nástrojom, ktorý umožňuje rozšírenú analýzu týchto dát. S pridaním dimenzií analyzovaných údajov a ich zobrazením v trojrozmernom priestore sa zvyšuje nielen pochopenie výsledkov experimentov, ale aj schopnosť interaktívne manipulovať s týmito dátami, čím sa získava oveľa hlbší pohľad na vykonané experimenty. Pre tento účel bol vyvinutý komponent NDMVR (N-DiMensional Virtual Reality), ktorý umožňuje vizualizáciu experimentálnych údajov vo virtuálnom prostredí.

Tento prístup prináša jednoznačnejšiu a interaktívnejšiu vizualizáciu komplexných experimentálnych údajov. Nástroje ako NDMVR umožňujú lepšie pochopenie a analýzu dát, ktoré by inak boli veľmi náročné na interpretáciu. Napriek tomu komponent NDMVR čelí viacerým výzvam, predovšetkým v oblasti používateľského zážitku (UX) a jeho prispôbitelnosti rôznym potrebám a preferenciám používateľov. Používateľské rozhranie v súčasnej verzii NDMVR a iných VR aplikácií nezohľadňuje dostatočne modularitu a flexibilitu, čo obmedzuje prispôbenie rozhrania podľa špecifických požiadaviek používateľov. Tento nedostatok prispôbitelnosti môže negatívne ovplyvniť efektivitu práce a celkovú používateľskú skúsenosť.

Problém je obzvlášť zjavný v prostredí A-Frame, frameworku na tvorbu VR aplikácií, ktorý síce poskytuje základné grafické primitíva a umožňuje jednoduchú implementáciu používateľských rozhraní v rôznych formách, ale často pôsobí amatérsky a neprofesionálne. Tvorba flexibilného rozhrania v tomto prostredí je buď neadekvátne očakávaniam používateľov v oblasti VR interakcie, alebo je čoraz náročnejšia a vyžaduje opakovanie definícií, čo negatívne ovplyvňuje čitateľnosť kódu a zvyšuje pracovnú náročnosť. Ďalším problémom je, že A-Frame poskytuje len hrubý základ pre zobrazenie vo VR, ktorý neposkytuje požadovanú špecifickú funkcionálnosť a dostatočnú flexibilitu pre komplexné aplikácie, aké sú

potrebné pre vizualizáciu vedeckých dát v reálnom čase, a nezohľadňuje špecifické požiadavky, ktoré komponent NDMVR musí splniť.

Aktuálny stav v oblasti vizualizácie dát a používateľských rozhraní vo VR aplikáciách pre vedecké účely, ako je NDMVR, vykazuje jasnú medzeru medzi potrebami používateľov a schopnosťami súčasných nástrojov. Tento problém si vyžaduje riešenie, ktoré by zlepšilo flexibilitu, prispôsobiteľnosť a celkový používateľský zážitok.

Cieľom tejto diplomovej práce je analyzovať súčasný stav komponentu NDMVR, identifikovať jeho nedostatky, najmä v oblasti používateľského rozhrania, a navrhnúť a implementovať vylepšenia, ktoré zlepšia jeho použiteľnosť, flexibilitu a estetickú kvalitu. Kľúčovou súčasťou práce bude aj podrobné hodnotenie používateľskej skúsenosti (UX) vo VR prostredí, ktoré bude zahŕňať analýzu a testovanie interakcie používateľov s aplikáciou. Hodnotenie UX bude zohľadňovať faktory ako intuitívnosť rozhrania, efektívnosť manipulácie s dátami, prispôsobiteľnosť rozhrania rôznym potrebám a preferenciám používateľov, a celkový komfort pri používaní. Na základe týchto zistení bude navrhnuté optimalizované používateľské rozhranie, ako aj zlepšenie celkového UX v prostredí, čo prispeje k vyššej kvalite a efektívnosti vizualizácie vedeckých dát.

Tento problém je stále relevantný, pretože súčasné nástroje neponúkajú dostatočnú flexibilitu a prispôsobiteľnosť, čo vedie k neefektívnemu využívaniu technologických možností, ktoré virtuálna realita ponúka. Riešenie tohto problému môže prispieť k významnému zlepšeniu výkonu a efektivity v oblasti vizualizácie vedeckých dát a prispieť k ďalšiemu vývoju nástrojov pre VR aplikácie v oblasti vedy. V neposlednom rade môže prispieť k inšpirácii pre ďalšie pokrytie medzier v spolupráci s metodikami hodnotenia, čo otvára priestor pre ďalšie vylepšenia a pokroky v tejto oblasti.

Formulácia úlohy

Prvým krokom bude analýza webového komponentu NDMVR z pohľadu použitých technológií a ich vplyvu na vizualizáciu údajov z fyzikálnych experimentov vo virtuálnej realite. Osobitná pozornosť sa bude venovať jeho aktuálnemu stavu, poskytovanému používateľskému rozhraniu a možnostiam jeho konfigurovateľnosti.

Následne sa vykoná prehľad existujúcich prístupov k hodnoteniu používateľskej skúsenosti (UX) vo virtuálnej realite s cieľom identifikovať relevantné metricky pre testovanie NDMVR. Výstupom tejto analýzy bude výber vhodnej meto-

díky testovania, ktorá umožní objektívne zhodnotiť prispôsobiteľnosť a použiteľnosť rozhrania i samotného komponentu.

Na základe výsledkov analýzy sa upraví komponent NDMVR tak, aby bol pripravený na testovanie. Dôraz sa bude klásť na čiastočnú automatizáciu testovacích procesov a zároveň na zachovanie možnosti jeho ďalšieho rozšírenia. Po implementácii úprav sa uskutoční samotné testovanie, získané dáta budú analyzované a výsledky vyhodnotené.

Na základe identifikovaných nedostatkov sa navrhnu a implementujú vylepšenia komponentu, pričom ich návrh a realizácia budú koordinované s ostatnými riešiteľmi projektu. Implementované zmeny budú reflektovať výsledky prvého testovania a v prípade potreby ďalej upravené, aby čo najlepšie odstránili zistené problémy.

Na záver sa zhodnotí efektivita realizovaných úprav. Výsledky poslúžia ako základ pre ďalší vývoj komponentu NDMVR.

1 Analytická časť - technológie

V tejto kapitole sa budeme venovať použitým technológiám a prístupom tvorby a interakcie s VR prostredím. Uvedieme stručný opis technológií, ako aj ich možnosti a prínosy ich použitia. Zároveň identifikujeme úlohy, ktoré tieto technológie budú plniť v implementačnej časti práce, pričom pri ich analýze budú spomenuté aj ich výhody a nevýhody v kontexte nášeho riešenia.

1.1 WebGL: Základ pre 3D grafiku v prehliadači

WebGL (Web Graphics Library) je knižnica JavaScriptu, ktorá umožňuje renderovanie 3D grafiky priamo v moderných webových prehliadačoch bez potreby inštalácie doplnkov alebo pluginov. WebGL poskytuje API (Application Programming Interface), čo je súbor pravidiel a protokolov, ktoré umožňujú rôznym softvérovým aplikáciám efektívne využívať grafický procesor (GPU) na zobrazenie grafiky [1]. To umožňuje tvorbu 3D aplikácií a vizualizácií priamo v prehliadači. WebGL je implementovaný v prehliadačoch ako natívna súčasť HTML5, čo znamená, že umožňuje interaktívne využitie 3D grafiky bez ďalších závislostí [2].

WebGL slúži ako základ pre pokročilé 3D knižnice a frameworky, ako je Three.js (pozri časť 1.2), ktorý abstrahuje zložitosť prác s nízkoúrovňovým API WebGL a poskytuje jednoduchšie nástroje pre vytváranie 3D scén [3].

1.2 Three.js: Jadro 3D vizualizácie

Three.js je open-source knižnica napísaná v jazyku JavaScript, ktorá umožňuje efektívnu tvorbu 3D grafiky priamo v internetovom prehliadači [3]. Využíva API WebGL, pričom nad ním poskytuje vysokú úroveň abstrakcie, čím eliminuje potrebu priamej práce so zložitými nízkoúrovňovými grafickými operáciami [2]. Táto vlastnosť robí knižnicu atraktívnou pre široké spektrum aplikácií vrátane interaktívnych vizualizácií, simulácií, hier, ako aj riešení v oblasti rozšírenej a vir-

tuálnej reality ¹.

Three.js ponúka rozsiahly súbor nástrojov na tvorbu realistických scén, vrátane podpory pokročilého osvetlenia, rôznych typov materiálov, tieňovania a animácií [2]. Vďaka tejto abstrakcii umožňuje efektívnu prácu s 3D objektmi bez potreby detailného poznania nízkoúrovňových grafických techník [3].

Dôležitú úlohu zohráva Three.js aj ako technologické jadro frameworku A-Frame (pozri časť 1.4), kde slúži ako základ pre vytváranie 3D a VR zážitkov. A-Frame využíva možnosti knižnice Three.js a nadväzuje na ne vrstvou ďalšej abstrakcie, umožňujúcou vývoj prostredníctvom jednoduchého deklaratívneho zápisu v jazyku HTML [4].

Práve táto vlastnosť z nej robí primárne API pre framework A-frame 1.4, kde slúži ako základná technológia pre vytváranie 3D a VR zážitkov. A-Frame je postavený na Three.js a poskytuje vrstvu abstrakcie a jednoduchý deklaratívny syntax pre vytváranie interaktívnych 3D scén pomocou HTML [4].

Jednou zo silných stránok knižnice Three.js je aktívna komunita vývojárov, ktorá zabezpečuje pravidelnú aktualizáciu, kvalitnú dokumentáciu a množstvo výučbových materiálov ². Knižnica je optimalizovaná pre rôzne typy zariadení vrátane smartfónov, tabletov a stolných počítačov, pričom dôraz na výkon zaručuje plynulé zobrazenie aj pri komplexných 3D scénach [3].

Okrem preddefinovaných entít frameworku A-Frame ponúka Three.js možnosť ich obchádzania a priamej manipulácie prostredníctvom vlastného API. Týmto spôsobom môžu vývojári vytvárať vlastné 3D objekty a presnejšie ich prispôbovať požiadavkám konkrétnej aplikácie ³.

1.3 WebXR: API pre virtuálnu a rozšírenú realitu na webe

WebXR (Web Extended Reality) ⁴ je aplikačné programové rozhranie (API), ktoré umožňuje vývojárom vytvárať interaktívne aplikácie pre virtuálnu realitu (VR) a rozšírenú realitu (AR) priamo v internetovom prehliadači. Poskytuje jednotné rozhranie na prácu s rôznymi VR a AR zariadeniami, ako sú VR headsety (napr. Oculus Rift alebo HTC Vive) a mobilné AR platformy [5].

WebXR je úzko integrované s technológiami ako Three.js (pozri časť 1.2) a

¹<https://threejs.org/>

²<https://threejs.org/docs/>

³<https://aframe.io/docs/1.7.0/introduction/developing-with-threejs>

⁴<https://immersiveweb.dev/>

A-Frame (pozri časť 1.4), čím zjednodušuje vývoj interaktívnych 3D aplikácií vo virtuálnej a rozšírenej realite. Three.js využíva WebGL (pozri časť 1.1) na renderovanie grafiky, zatiaľ čo WebXR zabezpečuje prepojenie s hardvérovými zariadeniami VR a AR [5]. Framework A-Frame, ktorý je postavený na Three.js, túto funkcionality rozširuje prostredníctvom deklaratívneho API, ktoré umožňuje rýchly vývoj VR aplikácií priamo v HTML prostredí [4].

Vďaka WebXR môžu tieto technológie fungovať naprieč širokým spektrom zariadení bez potreby individuálneho prispôbovania pre každé z nich. To výrazne uľahčuje vývoj multiplatformových riešení, ktoré poskytujú konzistentné a pohlcujúce používateľské zážitky ⁵.

1.4 A-Frame: Framework pre VR a 3D web aplikácie

A-Frame je open-source webový framework postavený na knižnici Three.js, ktorý výrazne uľahčuje tvorbu virtuálnych realít (VR) priamo v internetovom prehliadači. Využíva syntax podobnú HTML, vďaka čomu je prístupný aj vývojárom bez rozsiahlych skúseností s 3D grafikou [4]. Jeho architektúra je založená na entitovo-komponentovom systéme, čo umožňuje modulárny prístup a znovupoužiteľnosť prvkov ⁶.

A-Frame používa Three.js ako vykresľovací engine, avšak poskytuje vyššiu úroveň abstrakcie, ktorá zjednodušuje manipuláciu s 3D objektmi, materiálmi, osvetlením a animáciami [4]. Ako bolo uvedené v časti 1.2, A-Frame je možné rozšíriť o priame využitie Three.js kódu pre pokročilé úpravy a prispôbenia.

Framework je optimalizovaný pre široké spektrum zariadení, vrátane desktopov, mobilných zariadení a VR headsetov kompatibilných s technológiou WebXR [5]. Ponúka množstvo vstavaných funkcií ako geometrie, materiály, svetelné efekty, animácie, podpora 3D modelov či interaktivita prostredníctvom JavaScriptu [4]. Vďaka integrácii s WebXR umožňuje jednoduché prepojenie s rôznymi zariadeniami pre rozšírenú a virtuálnu realitu, čím poskytuje konzistentný a pohlcujúci používateľský zážitok naprieč platformami [5].

Ďalšou silnou stránkou frameworku je podpora interaktivity prostredníctvom mechanizmu raycastingu, ktorý umožňuje používateľom interagovať s objektmi v 3D priestore pomocou myši, dotykového rozhrania alebo VR ovládača ⁷.

A-Frame bol zvolený aj pre vývoj 3D a VR aplikácie v rámci riešenia tejto práce. Svoje využitie si našiel už v predchádzajúcich implementáciách uvedených v [6],

⁵<https://aframe.io/docs/1.7.0/introduction/vr-headsets-and-webxr-browsers>

⁶<https://aframe.io/docs/1.7.0/introduction/entity-component-system.html>

⁷<https://aframe.io/docs/1.7.0/introduction/raycaster>

[7], kde sa osvedčil ako flexibilné riešenie umožňujúce rýchlu iteráciu s minimálnymi nárokmi na objem kódu.

1.5 React: Základný prehľad a využitie v projekte

React je open-source JavaScriptová knižnica určená na tvorbu používateľských rozhraní, predovšetkým pre jednostránkové aplikácie (SPA) ⁸. Jeho hlavnou výhodou je efektívne reaktívne obnovovanie obsahu. Vďaka použitiu virtuálneho DOM-u React zabezpečuje, že sa aktualizujú iba tie časti rozhrania, ktoré sa reálne zmenili. Tento prístup výrazne zlepšuje výkonnosť aplikácií [8].

V projekte zameranom na 3D vizualizácie zohráva React dôležitú úlohu pri optimalizácii renderovania. Umožňuje priebežné a selektívne vykresľovanie len tých prvkov scény, ktoré si vyžadujú aktualizáciu. Tým sa znižuje výpočtová náročnosť, čo je obzvlášť dôležité pri komplexných 3D scénach, kde nie je efektívne renderovať celý obsah opakovane ⁹.

React zároveň umožňuje modulárnu štruktúru kódu rozdelenú na komponenty ¹⁰, ktoré sú znovupoužiteľné a nezávislé. Táto architektúra podporuje čistý a udržiavateľný vývoj. V kombinácii s rozšírením JSX, ktoré umožňuje vkladať HTML syntax priamo do JavaScriptu, vývojári dokážu vytvárať intuitívne a výkonné komponenty [8].

Na nižšej úrovni je možné do React aplikácie integrovať komponenty frameworku A-Frame (pozri časť 1.4), ktoré sa dynamicky renderujú ako súčasť 3D scény. Tento prístup zabezpečuje efektívne riadenie stavu a umožňuje postupnú kompozíciu 3D vizualizácií v reálnom čase ¹¹.

1.6 Využitie ROOTu vo vedeckej analýze

ROOT je objektovo-orientovaný softvérový rámec vyvinutý v CERN-e (Európskej organizácii pre jadrový výskum), primárne určený na analýzu rozsiahlych dátových súborov získaných z experimentov vo fyzike vysokých energií [9]. Bol navrhnutý s cieľom efektívne spracovávať, vizualizovať a analyzovať veľké objemy dát generovaných urýchľovačmi častíc, ako je napríklad experiment ALICE¹². Na

⁸<https://react.dev/>

⁹<https://react.dev/learn/render-and-commit>

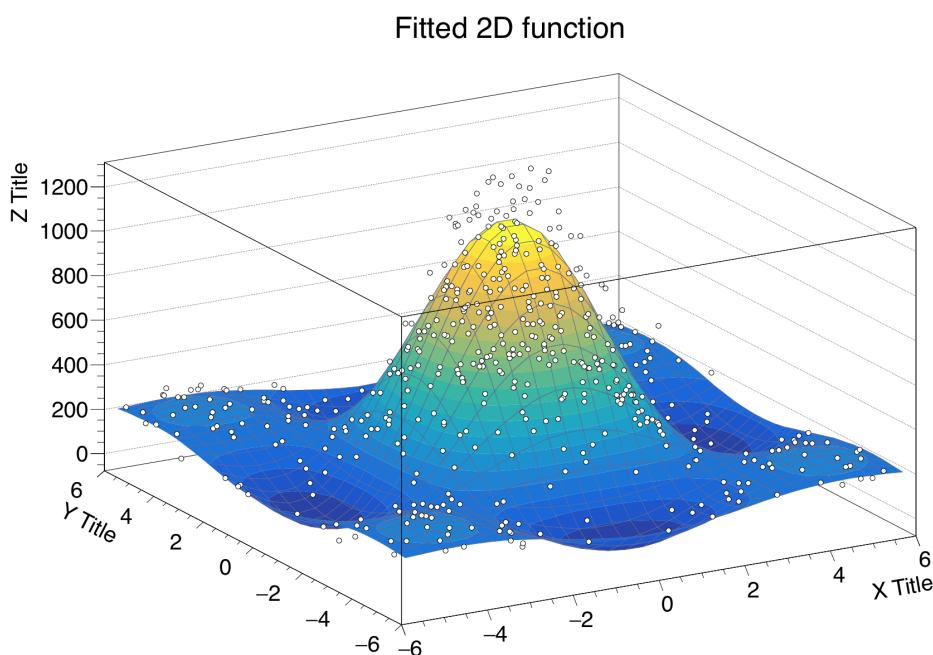
¹⁰<https://react.dev/reference/react/Component>

¹¹<https://www.npmjs.com/package/aframe-react>

¹²<https://root.cern/about/>

spracovaní týchto dát sa podieľa aj systém NDMVR podrobnejšie opísaný v kapitole 3.

ROOT je implementovaný v jazyku C++, čo umožňuje vysokú výkonnosť pri práci s údajmi [9] a zároveň poskytuje rozširiteľnosť prostredníctvom ďalších programovacích jazykov, ako napríklad Python, pomocou rozhrania PyROOT [10]. Tento rámec umožňuje vytváranie a správu zložitých dátových štruktúr, analýzu prostredníctvom štatistických a matematických nástrojov, modelovanie a fitovanie údajov, ako aj ich vizualizáciu pomocou interaktívnych grafov [11], čo je ilustrované na obr. 1.1.



Obr. 1.1: Príklad ROOT vizualizácie súboru údajov, prispôsobeného dvojrozmernou funkciou a zobrazeného ako farebný povrch. (zdroj: [12])

V oblasti fyzikálnej analýzy dát z urýchľovačov častíc sa ROOT využíva na spracovanie dát zaznamenaných detektormi, ktoré zachytávajú produkty zrážok častíc pri vysokých energiách [13]. ROOT umožňuje efektívne filtrovať signály od šumu, analyzovať korelácie medzi rôznymi fyzikálnymi veličinami a skúmať vlastnosti častíc, ako sú hmotnosť, náboj či spin [14]. Tieto analýzy môžu viesť k overeniu existujúcich fyzikálnych modelov alebo k objavom nových častíc a fenoménov, ako to bolo v prípade objavu Higgsovho bozónu v experimentoch ATLAS a CMS [15].

Medzi kľúčové nástroje ROOT-u patrí možnosť vytvárania histogramov, prispôbovanie distribúcií (fitovanie), správa veľkých datasetov vo formáte ROOT file a využitie stromových štruktúr (TTree) na efektívne spracovanie a analýzu

dát [9]. ROOT sa stal neoddeliteľnou súčasťou spracovania dát v experimentoch na LHC a je široko používaný vo vedeckej komunite ¹³.

1.6.1 JSRoot: Webová vizualizácia ROOT dát

JSRoot je JavaScriptová knižnica, ktorá umožňuje interaktívnu vizualizáciu ROOT dát priamo vo webových prehliadačoch, bez potreby inštalácie ROOT softvéru. Podporuje renderovanie histogramov, grafov a 3D objektov [16].

V komponente NDMVR, podrobnejšie opísanom v kapitole 3, ktorý funguje prostredníctvom webového rozhrania, sú využívané frameworky React a A-Frame v kombinácii s knižnicou JSRoot na vizualizáciu a analýzu ROOT dát priamo v prehliadači [6]. React poskytuje komponentovú štruktúru pre tvorbu užívateľského rozhrania [8], A-Frame umožňuje 3D vizualizácie [17] a JSRoot umožňuje prácu s ROOT dátami bez potreby spúšťania ROOT softvéru na lokálnom počítači [16].

Integrácia JSRoot s Reactom a A-Frame umožňuje vytvárať komplexné 3D vizualizačné nástroje, ktoré sú prístupné priamo cez webové prehliadače. Jedným z príkladov je react-3d-editor ¹⁴, ktorý demonštruje použitie A-Frame v kombinácii s Reactom na tvorbu 3D vizualizačných nástrojov.

Táto integrácia poskytuje efektívny spôsob, ako prezentovať a analyzovať veľké objemy dát vo fyzike vysokých energií prostredníctvom interaktívnych webových aplikácií.

¹³<https://root.cern/>

¹⁴<https://github.com/salgum1114/react-3d-editor>

2 Analýza prístupov a aspektov merania používateľského zážitku

V tejto časti by sme chceli diskutovať o aspektoch hodnotenia UX, štandardizovaných metrikách hodnotenia, odporúčanej forme takéhoto hodnotenia a v neposlednom rade, možných výsledkoch, ktoré dané hodnotenie môže priniesť alebo odhaliť.

2.1 Klasifikácia UX testovania

Prvou otázkou, ako aj analyzovaným aspektom je, ktoré prvky je dôležité merať a aké potenciálne roviny by malo UX testovanie sledovať, odzrkadľovať.

Práca [18] sa zaoberá práve touto otázkou. Okrýva pohľad na sledované roviny testovania, ako aj rôzne metódy merania použiteľnosti a používateľskej skúsenosti (UX), s dôrazom na ich význam pri hodnotení a vývoji produktov. Základný koncept merania je v tomto kontexte podľa normy ISO 9241-11, ktorá hovorí o schopnosti produktu dosiahnuť stanovené ciele s efektívnosťou, efektivitou a spokojnosťou, predstavený ako **Použiteľnosť**.

Práca [18] popisuje kategorizáciu metód, ako aj aspektov hodnotenia na sumatívne a formatívne. Sleduje ich úlohu pri posudzovaní použiteľnosti v rôznych fázach návrhu a vývoja produktov.

Sumatívne merania sú kľúčové pre stanovenie základnej úrovne použiteľnosti, porovnávanie produktov a hodnotenie dodržiavania požiadaviek na použiteľnosť. Tieto merania vyžadujú platnosť a spoľahlivosť, aby poskytovali zmysluplné poznatky, ktoré sa získavajú z reprezentatívnych vzoriek užívateľov vykonávajúcich úlohy v realistických kontextoch. Daný text práce [18] zdôrazňuje dôležitosť použitia štandardizovaných metrík, pre adekvátne kvantitatívne porovnanie.

Formatívne hodnotenie dopĺňa sumatívne prístupy identifikáciou problémov s použiteľnosťou v raných fázach návrhu a jemným doladením používateľ-

ských požiadaviek. Formatívne hodnotenie kvalitatívneho charakteru poskytuje cenné poznatky, ktoré napomáhajú iteratívnemu zlepšovaniu dizajnu. V praxi kombinácia formatívneho a summatívneho prístupu zvyšuje efektivitu návrhového procesu.

Dôležitosť merania UX a použiteľnosti spočíva v komplexnom pochopení potrieb používateľov a v zlepšovaní kvality produktov s cieľom poskytnúť výborné používateľské skúsenosti.[18] Práve pre lepšie pochopenie potrieb nám práca [18] ukázala zaujímavý, rozšírený pohľad na použiteľnosť a to na úrovni systému, ktorý by mal zahŕňať schopnosť **ľahkého učenia, dostupnosti a bezpečnosti pri dosahovaní potrieb používateľa**. Tieto aspekty spoločne prispievajú k celkovému používateľskému zážitku.

2.2 Výber sledovaných oblastí UX testovania

Pri hodnotení používateľskej skúsenosti (UX) je z nášho pohľadu kľúčové rozlíšiť jednotlivé aspekty, ktoré ovplyvňujú celkovú spokojnosť používateľov vzhľadom na kategóriu riešenia a jeho určenie. Medzi tri hlavné dimenzie nášho záujmu, na ktoré by sme chceli poskytnúť odpoveď v tejto časti, patria: Aké vlastnosti sú dôležitejšie z pohľadu skúmania a hodnotenia UX — hedonické, alebo pragmatické? Ktoré UX faktory sú najdôležitejšie na sledovanie pri hodnotení UX? A aké vlastnosti je nutné zahrnúť z pohľadu konkrétnej kategórie systému?

Odpoveď na tieto otázky nám poskytla práca [19]. Táto práca detailne skúma význam pragmatických a hedonických kvalít v UX a ich vplyv na používanie digitálnych nástrojov v pracovnom aj voľnočasovom prostredí. Zameriava sa na dve hlavné kategórie UX kvalít — pragmatické a hedonické — a zaoberá sa aj otázkou, v akých prípadoch každá z nich zohráva odlišnú úlohu v závislosti od kontextu použitia. Pre správne nastavenie UX testovania je dôležité analyzovať jednotlivé body tejto práce a jej výsledky, aby sme pochopili ich vplyv na celkové hodnotenie, ako aj na samotné digitálne produkty v závislosti od ich kategorizácie a určenia.

2.2.1 Sledované dimenzie UX testovania a odpovede skúmania

Výskum [19] identifikoval tri dimenzie, na ktoré sa snažíme poskytnúť odpovede. Prvou dimenziou bola identifikácia dôležitosti z pohľadu skúmania a hodnotenia UX, konkrétne hedonických a pragmatických vlastností. V štúdii [19] mali účastníci vybrať UX faktory z UEQ+ (User Experience Questionnaire Plus), ktoré považovali za kľúčové pri hodnotení používateľskej skúsenosti kolaboračných ná-

strojov. Týmto smerom by sa mala aj naša práca v budúcnosti uberať, konkrétne ide o komponentu NDMVR, podrobnejšie opísanú v kapitole 3. Účastníci mali tiež možnosť navrhnúť ďalšie faktory, ktoré neboli zahrnuté v zozname UEQ+. Tieto faktory boli následne zhrnuté a zoradené podľa dôležitosti prostredníctvom diskusie na webovej konferencii medzi účastníkmi. Štúdia identifikovala 18 kľúčových UX faktorov, ktoré ovplyvňujú používateľskú skúsenosť v profesionálnych a voľnočasových nástrojoch.

Odpoveďou je podľa [19] potvrdenie, že rôzne UX aspekty majú rôznu váhu v závislosti od kontextu použitia. V profesionálnych nástrojoch dominujú pragmatické kvality, zatiaľ čo pri voľnočasových aplikáciách zohrávajú väčšiu úlohu hedonické faktory. Napriek tomu, v závere práce je spomenuté možné mierne nevyváženie týchto vlastností v niektorých špecifických oblastiach, no úspešný dizajn UX naprieč rôznymi kontextami systémov nevyhnutne vyžaduje zohľadnenie oboch skupín faktorov. Takisto je potvrdené, že vybraných 18 kľúčových UX faktorov je rovnomerne zastúpených v oboch kategóriách: pragmatických a hedonických faktoroch.

Pragmatické faktory: Tieto faktory sa týkajú funkcionality, efektívnosti a spoľahlivosti systému [19]. Podľa analyzovanej štúdie [19] medzi kľúčové pragmatické faktory, ktoré poskytujú odpoveď na nasledujúce otázky, patria:

- **Prístupnosť:** Umožňuje produkt využívanie čo najväčšiemu počtu používateľov bez obmedzení?
- **Adaptabilita:** Môže byť produkt prispôsobený osobným preferenciám alebo osobným pracovným štýlom?
- **Spoľahlivosť:** Cíti používateľ kontrolu nad interakciou? Je to bezpečné a predvídateľné?
- **Efektívnosť:** Môžu používatelia vyriešiť svoje úlohy bez zbytočného úsilia? Reaguje produkt rýchlo?
- **Funkčnosť:** Ponúka produkt adekvátny súbor funkcií na úspešné dokončenie úloh?
- **Interakcia:** Podporuje produkt interakciu medzi používateľmi?
- **Intuitívne:** Môže byť produkt použitý okamžite bez potreby školenia alebo pomoci?
- **Prezieravosť:** Je jednoduché zoznámiť sa s produktom a naučiť sa ho používať?

- **Užitočnosť:** Prináša používanie produktu výhody?

Hedonické faktory: Tieto faktory súvisia s emocionálnymi a estetickými aspektmi UX [19]. Štúdia [19] identifikovala nasledovné hedonické faktory, ktoré poskytujú odpoveď na nasledujúce otázky:

- **Estetika:** Vyzerá produkt príťažlivo?
- **Pripojenie:** Je možné vytvoriť spojenie s inými jednotlivcami prostredníctvom používania produktu?
- **Atraktivita:** Celkový dojem z produktu. Páči sa používateľom alebo nie?
- **Jasnosť:** Aký je dojem z poriadku, štruktúry a vizuálnej komplexnosti grafického používateľského rozhrania?
- **Novosť:** Je dizajn produktu kreatívny? Zaujme používateľov?
- **Sebaprezentácia:** Je jednoduché prezentovať sa prostredníctvom produktu?
- **Stimulácia:** Je používanie produktu vzrušujúce a motivujúce? Je používanie zábavné?
- **Dôvera:** Sú dáta používateľa v bezpečí a nebudú zneužitú na jeho úkor?
- **Hodnota:** Vyzerá dizajn produktu profesionálne a odráža vysokú kvalitu?

Štúdia [19] zároveň poskytuje odpoveď na druhú dimenziu, ktorú sme skúmali, konkrétne: Ktoré UX faktory sú najdôležitejšie pri hodnotení používateľskej skúsenosti? Odpoveď prináša práca v podobe hodnotenia dôležitosti jednotlivých UX faktorov. Účastníci boli, na základe zoznamu faktorov vybraných v predchádzajúcej fáze, požiadaní, aby ohodnotili dôležitosť týchto faktorov pre vybrané produkty pomocou sedembodovej Likertovej škály. Účastníci si mali vybrať produkty z nasledujúcich kategórií: kolaboračné platformy, systémy správy dokumentov, instant messaging služby, systémy správy úloh a webové konferenčné systémy.

Tretia dimenzia sa zameriava na určenie rozdielov medzi kontextami systémov a ich hodnoteniami. V poslednej fáze štúdie [19] bol použitý dvojstranný t-test na identifikáciu rozdielov v dôležitosti UX faktorov medzi produktovými skupinami a kontextami použitia. Cieľom tejto analýzy bolo zistiť, ktoré UX faktory vykazujú významné rozdiely v hodnoteniach pri použití produktov v profesionálnom a voľnočasovom prostredí.

Poskytnuté odpovede na otázky týkajúce sa skúmaných dimenzií 2 a 3 boli nasledovné. Dôležitosť UX faktorov sa líši v závislosti od kontextu použitia, konkrétne v prípade kolaboračných nástrojov. K týmto nástrojom by sa v budúcnosti mal pridať aj NDMVR. Autori práce [19] dospeli k záveru, že v profesionálnych nástrojoch majú väčšiu váhu pragmatické faktory, ako spoľahlivosť a efektívnosť, zatiaľ čo v prípade voľnočasových aplikácií sú významnejšie hedonické faktory, ako novosť a stimulácia. Napriek tomu práca [19] ukazuje existenciu niekoľkých univerzálne dôležitých faktorov opísaných v časti 2.2.2, ktoré sú relevantné pre úspešný dizajn UX naprieč rôznymi kontextami systémov.

2.2.2 Konkretizácia UX faktorov na základe skúmaných dimenzií

Na základe nadobudnutých odpovedí sme sa rozhodli pre univerzálne riešenie, spomenuté v závere práce [19], a to s takmer vyváženým zastúpením pragmatických aj hedonických faktorov. Kategorizácia systému bude vnímaná ako kategória v práci [19], ktorá sa zaoberá úspešným dizajnom UX naprieč rôznymi kontextami systémov. Záverom týchto pozorovaní je súhrn týchto univerzálnych odporúčaných faktorov, ktoré sú nevyhnutné pri hodnotení UX. Ide o redukciu 18 UX faktorov na 7, ako je v hodnotení práce [19] uvedené. Existuje skupina faktorov UX, ktorých význam zostáva rovnaký vo všetkých kontextoch a pre všetky typy nástrojov. Patria sem:

- **Prístupnosť**
- **Adaptabilita**
- **Jasnosť**
- **Interakcia**
- **Intuitívne použitie**
- **Zrozumiteľnosť**
- **Sebaprezentácia**

Tieto faktory sa zdajú byť univerzálne a nezávislé od konkrétneho použitia.

2.2.3 Faktory kvality systému

Napriek zisteným záverom v časti 2.2.2, sme sa rozhodli poskytnúť ešte jeden pohľad na špecifikáciu konkrétnych črt merania UX pre kvalitný systém. Konkrétne črty zmysluplného a validného UX merania nám poskytla práca [18], zameraná na analýzu zmysluplných metrík, a to konkrétne na platné a užitočné meranie používateľskej skúsenosti.

Práca [18] identifikuje charakteristické črty kvalitného systému, ktoré sú relevantné pre oblasti merania UX, ako napríklad:

- **UX (Používateľská skúsenosť):** Zahŕňa všetky aspekty používateľskej skúsenosti pri interakcii s produktom, službou, prostredím alebo zariadením. Je zjednotením pragmatických a hedonických cieľov používateľa.
- **Funkčnosť:** Schopnosť produktu vykonávať požadované funkcie a efektívne a spoľahlivo dosahovať ciele používateľa.
- **Použitelnosť používateľského rozhrania:** Jednoduchosť a efektívnosť, s akou používatelia môžu produkt používať.
- **Zrozumiteľnosť:** Ako rýchlo a ľahko sa môžu používatelia naučiť používať produkt. Schopnosť produktu byť použiteľný pre používateľov s rôznymi schopnosťami a v rôznych kontextoch.
- **Bezpečnosť:** Minimalizácia rizika nežiaducich následkov pri používaní produktu.
- **Estetické atribúty:** Vizuálna a zmyslová atraktivita produktu.
- **Vhodné funkcie:** Implementácia funkcií, ktoré odpovedajú potrebám používateľov.
- **Dizajn užívateľského rozhrania:** Dizajn, ktorý zjednodušuje používanie produktu.
- **Technická dostupnosť a jednoduchosť použitia:** Systém by mal byť prístupný a jednoduchý na používanie pre všetkých používateľov, pričom je zabezpečený a bezpečný.
- **Dosahovanie pragmatických a hedonických cieľov UX:** Efektivita a efektívnosť pri vykonávaní úloh, radosť, potešenie a emocionálne reakcie pri používaní produktu. Skutočná používateľská skúsenosť (UX).

- **Estetické atribúty:** Miera spokojnosti používateľa pri dosahovaní pragmatických a hedonických cieľov, reakcie založené na zmyslovom vnímaní produktu (potešenie).
- **Dôvera:** Úroveň spokojnosti používateľa, že produkt bude správne fungovať a vyhovovať jeho potrebám.

Tieto črty sme sa rozhodli použiť ako vzor správnosti a teda ako cieľ nášho sledovania pri meraní UX. Výsledky budú slúžiť ako doplnenie k výsledkom poskytnutým v časti 2.2.2.

2.2.4 Prepojenie univerzálnych faktorov s faktormi kvality systému

Časť 2.2.2 poskytuje vstup v podobe univerzálnych pragmatických a hedonických faktorov, teda cieľov, ktoré je potrebné naplniť pre úspešný dizajn UX naprieč rôznymi kontextami systémov. Tieto faktory sa pokúsime nadviazať v podobe prieniku výsledkov z častí 2.2.2 a 2.2.3 do zmysluplného výstupu v podobe kategórií, ktoré budú odzrkadľovať sledované oblasti merania UX tejto práce. Neskôr sa budeme snažiť identifikovať a aplikovať konkrétnu relevantnú metriku pre tieto oblasti.

1. Prístupnosť $\iff$ Technická dostupnosť a jednoduchosť použitia

Význam: Tieto dva faktory kategorizujeme ako jednoduchosť použitia prostredníctvom dobrej prístupnosti produktu pre väčšinu používateľov. V tejto práci bude prístupnosť a technická dostupnosť považovaná za nevyhnutnú pre dosiahnutie základnej používateľskej skúsenosti, pričom ideálnym cieľom bude, aby bol produkt jednoduchý na použitie pre rôzne skupiny používateľov.

2. Adaptabilita $\iff$ Vhodné funkcie

Význam: Táto kategória je charakterizovaná ako schopnosť systému prispôbiť sa rôznym potrebám používateľov. Pre tento účel budeme sledovať, neskôr v implementácii aj definovať vhodné funkcie, teda funkcie, ktoré sa musia zosúladiť s týmto aspektom, aby produkt mohol vyhovieť rôznym preferenciám a potrebám, ideálne dokonca aj dynamicky sa meniacim. Tieto faktory sú sledované, keďže zohrávajú kľúčovú úlohu nielen pri hodnotení, ale aj pri samotnom návrhu implementovaných zmien.

3. Jasnosť $\iff$ Dizajn užívateľského rozhrania

Význam: Jasnosť dizajnu a jeho zrozumiteľnosť budeme v tejto kategórii vnímať ako meranú rýchlosť pochopenia, prípadne krivku učenia sa, ako používať produkt. Zároveň budeme predovšetkým sledovať používateľské rozhranie, ktoré má byť prehľadné, intuitívne a jednoduché, čo pomáha používateľovi v orientácii a minimalizuje zbytočné zmätky.

4. Interakcia $\iff$ Použiteľnosť používateľského rozhrania

Význam: Kategória predstavuje názor, kde dobre navrhnuté interakcie medzi používateľom a systémom sú kľúčové pre efektívnosť používateľského rozhrania. Interakcia ovplyvňuje, ako používateľ komunikuje s produktom, čo priamo súvisí s jeho použiteľnosťou. Preto budeme sledovať meniacu sa použiteľnosť systému.

5. Intuitívne použitie $\iff$ Zrozumiteľnosť

Význam: Intuitívne použitie pre nás bude znamenať, že produkt je ľahko pochopiteľný a používatelia sa môžu rýchlo naučiť jeho používanie bez potreby zložitého zaškolenia. Zrozumiteľnosť produktu súvisí s rýchlosťou, akou sa používateľ môže naučiť používať produkt bez zbytočných ťažkostí. Toto sledovanie kategorizujeme tiež ako vnímanú použiteľnosť zo strany používateľov, prípadne taktiež sledovanie krivky učenia sa.

6. Zrozumiteľnosť $\iff$ Zrozumiteľnosť

Význam: Zrozumiteľnosť produktu a jeho ľahká použiteľnosť budú kľúčovými faktormi v oblasti učenia sa a venovania pozornosti systému. Ak je produkt jasný a intuitívny, používateľ sa rýchlo zorientuje a získa pozitívnu skúsenosť, ktorá sa odzrkadlí na jeho pozornosti a záujme, nakoľko predpokladáme, že v opačnom prípade by mohla jeho pozornosť klesať, ako je tomu aj v procese učenia sa podľa práce [20].

7. Sebaprezentácia $\iff$ Estetické atribúty

Význam: Sebaprezentácia v kontexte používania produktu sa týka toho, ako používateľ vníma sám seba prostredníctvom interakcie s produktom, teda ako sa prezentuje jeho používaním [19]. V tejto kategórii predpokladáme, že ak produkt podporuje pozitívny obraz používateľa prostredníctvom svojho dizajnu a estetických atribútov, môže prispieť k naplneniu cieľa sebaprezentácie. Tento faktor sledujeme, nakoľko podľa [18] estetické prvky ovplyvňujú emocionálnu reakciu používateľa a môžu pozitívne formovať jeho interakciu s produktom.

2.2.5 Spoločná skupina meraných faktorov

Sledované faktory po nájdení prieniku množín kvalít a hedonických či pragmatických cieľov identifikujeme ako: **Prístupnosť, Adaptabilita, Jasnosť, Interakcia, Intuitívnosť, Zrozumiteľnosť, Estetika**

2.3 Rady a odporúčaná forma sledovania UX vo VR systémoch

Podkapitola analyzuje konkrétne formy merania UX v prostredí virtuálnej reality, zohľadňuje odporúčania a rady. Cieľom je vybrať optimálnu formu testovania vzhľadom na požiadavky ako čas, spektrum respondentov, schopnosť adekvátne zachytiť výsledky a ich prezentáciu. Na úvod sa zameriame na možnosti zberu dát a realizovateľné formy testovania.

2.3.1 Odporúčaná forma testovania UX vo VR systémoch

Štúdia [21] sa zameriava na metódy hodnotenia užívateľskej skúsenosti vo virtuálnej realite. Autori analyzovali 364 článkov a identifikovali, že najčastejšie používanými metódami sú dotazníky a rozhovory. Zároveň navrhujú aj ďalšie hodnotiace metódy, ktoré zatiaľ neboli v štúdiách VR využité. Odhalené boli obmedzenia tradičných princípov dizajnu v oblasti HCI (Human-Computer Interaction) pri vývoji a testovaní VR systémov.

Autori práce [21] tvrdia, že HCI nepokrýva niektoré kľúčové aspekty zážitku z VR, ako sú:

- **Prítomnosť** – pocit ponorenia sa do virtuálneho prostredia
- **Manipulácia s objektmi** – spôsob, akým používateľ interaguje s virtuálnymi objektmi
- **Vnímanie** – spôsob, akým užívateľ spracováva informácie v prostredí VR

Keďže autori práce [21] sa domnievajú, že tieto faktory zohrávajú zásadnú úlohu pri tvorbe používateľskej skúsenosti (UX) vo VR, ich efektívne hodnotenie umožňuje identifikovať a odstrániť nedostatky v dizajne VR systémov.

Cieľom výskumu bolo vytvorenie stručného sprievodcu metódami hodnotenia UX vo VR, pričom poskytli odporúčania vzhľadom na formu testovania. Opísané boli rôzne metódy hodnotenia UX, vrátane ich výhod, nevýhod a požiadaviek na vybavenie. Práca [21] pritom rozdeľuje metódy do dvoch kategórií:

- **Postojové metódy** – napr. dotazníky a rozhovory, ktoré sa spoliehajú na subjektívnu spätnú väzbu používateľov.
- **Behaviorálne metódy** – napr. sledovanie pohybov a fyziologické merania, ktoré sa zameriavajú na objektívne dáta získané z pozorovania správania používateľov.

Významným záverom bolo určenie najbežnejšie používaných metód – dotazníkov a rozhovorov, nakoľko sú lacné a ľahko implementovateľné.

Postojové metódy majú obmedzenia, pretože sú závislé od kvality spätnej väzby užívateľov, ktorí môžu otázky nesprávne pochopiť, neúplne opísať svoj zážitok alebo neúmyselne skresliť odpovede [21]. Preto boli označené za nie úplne optimálnu voľbu v prípade ich samostatnej implementácie. Behaviorálne metódy poskytujú presnejšie dáta, keďže sú založené na výkonnostných a fyziologických meraniach. Odporúčaním bolo kombinovanie metód, ktoré však ešte neboli dôkladne testované. To otvára priestor pre ďalší výskum, najmä v oblasti automatizovaného behaviorálneho výskumu. Práca [21] teda zdôrazňuje výhody kombinácie oboch prístupov pre presnejšie hodnotenie UX.

Výsledkom analyzovanej práce [21] sa stal najčastejšie používaný a overený spôsob získania adekvátnych výsledkov prostredníctvom post-hodnotenia testovania dotazníkom, doplneným o názory respondentov, a zberu dát získaných počas testovania na základe ich akcií a správania. Ako metódu zberu údajov sme zvolili meranie času vykonania úlohy, keďže ide o výkonnostnú metriku poskytujúcu kvantifikovateľné údaje o efektivite interakcie používateľa s VR systémom. Tento parameter analyzujeme ako ukazovateľ efektivity interakcie aj ako objektívne údaje získané z pozorovania správania používateľov, čím slúži ako behaviorálna metóda. Príkladom je štúdia [22], ktorá hodnotila kognitívny výkon mladých dospelých pri vykonávaní Trail Making Testu vo virtuálnej realite, pričom medzi hlavné metriky patrili čas dokončenia úlohy a presnosť.

2.3.2 UX rady a konvencie pre VR systémy a adekvatne poskytnutie, sledovanie UX

V oblasti manipulácie a dizajnu používateľského rozhrania VR aplikácií chýbajú štandardy a konvencie, čo podčiarkuje dôležitosť používateľského testovania [23].

Podľa Nielsena [23] je kľúčové uľahčiť používateľom prácu v novom rozmeri VR, ktorý nie je bežný, nakoľko VR prináša nové možnosti, ktoré nie sú tak prirodzené. Dôležité je zabezpečiť, aby používanie bolo čo najviac intuitívne a prirodzené, aby sa používatelia viac zapájali do tejto technológie. Dôležitým aspektom

je jednoduché ovládanie, intuitívne rozhranie a dostupné návody, ktoré umožnia používateľom komfortne využívať nové možnosti, ktoré VR ponúka.

Pri získavaní spätnej väzby je dôležité zadávať používateľom konkrétne úlohy, ako je vyhľadanie, objavenie alebo využitie určitého prvku. Ukážky demo verzií a voľného skúšania nie sú postačujúce. Úlohy by mali byť zamerané na reálne situácie a mali by zohľadňovať očakávané reakcie [23].

Získavanie spätnej väzby, napr. vo forme dotazníka je efektívnejšie realizovať priamo v prostredí VR, aby sa predišlo rušivosti spojenej s odobratím headsetu a vyplnením dotazníka mimo VR prostredia. Takáto manipulácia s headsetom môže ovplyvniť používateľské skúsenosti a viesť k skresleným výsledkom [23].

2.4 Summatívne meranie v praxi: Efektívne hodnotenie použiteľnosti

Analyzovaným aspektom tejto časti je summatívne meranie kritérií zameraných na hodnotenie použiteľnosti, ktoré môžeme hodnotiť prostredníctvom dotazníka System Usability Scale (SUS). Tento testovací štandard predstavil John Brooke už v 80. rokoch minulého storočia [24]. Na základe analýzy [24] a [23] sme sa rozhodli pre túto jednoduchú, overenú a rýchlu metódu hodnotenia použiteľnosti, ktorá umožňuje sledovanie meraných faktorov systému. Aj keď SUS neposkytuje explicitné hodnotenie všetkých špecifických faktorov uvedených v časti 2.2.2, niektoré z týchto faktorov môžu byť čiastočne nepriamo zahrnuté do hodnotenia SUS vďaka celkovému vnímaniu použiteľnosti [24].

2.4.1 Dotazník System Usability Scale (SUS)

Dotazník System Usability Scale (SUS) slúži na meranie vnímanej použiteľnosti a umožňuje získať adekvátne odpovede na otázky týkajúce sa užitočnosti systému [24]. Je jedným z najznámejších dotazníkov používaných vo výskume UX [24]. Informácie o jeho častom použití a kategorizácií ako svetovo uznávaný testovací štandard [24] nám výrazne pomohli pri jeho výbere. SUS je post-testovací nástroj, ktorý je poskytnutý účastníkom po skončení celej relácie testovania použiteľnosti. Obsahuje 10 otázok na Likertovej škále [23] a je znázornený na obr. 2.1.

Účastníci hodnotia svoj súhlas s každým tvrdením týkajúcim sa produktu alebo funkcie v teste na stupnici od 1 do 5. Hodnota 1 znamená úplný nesúhlas, zatiaľ čo hodnota 5 znamená, že účastník úplne súhlasí s daným tvrdením. SUS dotazník poskytuje skóre v rozmedzí od 0 do 100, ktoré reprezentuje použiteľnosť

[23]. Dôležitým faktom je, že pomocou SUS neidentifikujeme konkrétne problémy, pretože ide o kvantitatívne hodnotenie, ktoré poskytuje základ pre zistenie celkovej použiteľnosti produktu [24].

Q1:	I think that I would like to use this system frequently.
Q2:	I found the system unnecessarily complex.
Q3:	I thought the system was easy to use.
Q4:	I think that I would need the support of a technical person to be able to use this system.
Q5:	I found the various functions in this system were well integrated.
Q6:	I thought there was too much inconsistency in this system.
Q7:	I would imagine that most people would learn to use this system very quickly.
Q8:	I found the system very cumbersome to use.
Q9:	I felt very confident using the system.
Q10:	I needed to learn a lot of things before I could get going with this system.

Obr. 2.1: SUS (zdroj: [23])

Podľa Jeffa Saura, ktorý uskutočnil rozsiahle porovnanie skóre System Usability Scale (SUS) na rôznych systémoch, je 68 percento považované za minimálne skóre SUS, ktoré možno kategorizovať ako dobré [23]. Pri vývoji webových stránok je potrebné dosiahnuť skóre 80 alebo vyššie, aby sme sa umiestnili v top 10 percentách z hľadiska použiteľnosti, zatiaľ čo skóre 73 nám zaručí len umiestnenie v top 30-tich percentách [24].

2.5 Sumatívne meranie v praxi: Neobvyklé prístupy k hodnoteniu angažovanosti

Aspekt sumatívneho hodnotenia, konkrétne meranie úrovne zapojenia používateľa, teda jeho angažovanosti, je podrobne skúmaný v štúdiu [25], ktorá sa zameriava na meranie angažovanosti používateľov v digitálnych technológiách pomo-

cou User Engagement Scale (UES). V rámci tejto štúdie bola metodika merania angažovanosti UES významne zdokonalená a skrátená. Autori štúdie navrhli a testovali nové, presnejšie verzie dotazníkov, ktoré sme sa rozhodli použiť na meranie angažovanosti používateľov vo virtuálnej realite (VR). Toto meranie umožňuje sledovanie vybraných UX faktorov systému uvedených v časti 2.2.2 prostredníctvom zahrnutia hlavných faktorov angažovanosti.

2.5.1 Refined User Engagement Scale (UES-LF)

Rafinovaná verzia UES bola vyvinutá na základe pôvodnej verzie, ktorá pozostávala z 31 položiek rozdelených do 6 dimenzií: estetická príťažlivosť, sústredená pozornosť a vnímaná použiteľnosť, odolnosť, novosť a pocit angažovanosti. V rafinovanej verzii dotazníka UES sa dimenzie odolnosti, novosti a pocitu angažovanosti zlúčili do štvrtého faktora, ktorým je Faktor odmeny (RW).

Každá položka je hodnotená na 5-bodovej škále od "rozhodne nesúhlasím" po "rozhodne súhlasím". Počet otázok zostáva rovnaký. Výsledky ukázali, že nová vylepšená verzia UES má vyššiu spoľahlivosť a presnosť ako pôvodná verzia [25].

2.5.2 UES Short Form (UES-SF)

Štúdia [25] taktiež odhaľuje novú formu UES v podobe krátkej verzie, ktorá bola vytvorená na základe rafinovanej verzie a je určená na rýchle získanie prehľadu o angažovanosti používateľov s produktom. Výsledky ukázali, že táto verzia má vysokú spoľahlivosť a presnosť, pričom zachováva výsledky metódy UES-LF [25].

Na obr. 2.2 je znázornených 12 otázok, ktoré tvoria novú formu sledovaných faktorov.

1. **Estetická príťažlivosť (AE)** – 3 otázky
2. **Sústredená pozornosť (FA)** – 3 otázky
3. **Vnímaná použiteľnosť (PU)** – 3 otázky
4. **Faktor odmeny (RW)** – 3 otázky

2.5.3 Faktory angažovanosti

UES metodika hodnotenia identifikuje štyri hlavné faktory angažovanosti používateľov. Tieto faktory poskytujú kvantitatívne hodnotenie prostredníctvom škály od 1 do 5.

Ako uvádza práca [25], definícia týchto faktorov je nasledovná:

FA-S.1:	I lost myself in this experience.
FA-S.2:	The time I spent using Application X just slipped away.
FA-S.3:	I was absorbed in this experience.
PU-S.1:	I felt frustrated while using this Application X.
PU-S.2:	I found this Application X confusing to use.
PU-S.3:	Using this Application X was taxing.
AE-S.1:	This Application X was attractive.
AE-S.2:	This Application X was aesthetically appealing.
AE-S.3:	This Application X appealed to my senses.
RW-S.1:	Using Application X was worthwhile.
RW-S.2:	My experience was rewarding.
RW-S.3:	I felt interested in this experience.

Obr. 2.2: UES-SF (zdroj: [25])

1. **Estetická príťažlivosť (Aesthetic Appeal, AE):** Tento faktor odráža vizuálnu príťažlivosť a estetické vlastnosti produktu alebo systému, ktoré môžu ovplyvniť pozitívne vnímanie používateľov.
2. **Sústredená pozornosť (Focused Attention, FA):** Tento faktor sa týka schopnosti systému alebo produktu udržať pozornosť používateľa počas interakcie, čo môže priamo ovplyvniť jeho angažovanosť.
3. **Vnímaná použiteľnosť (Perceived Usability, PU):** Tento faktor hodnotí, ako efektívne a bezproblémovo môže používateľ vykonávať úlohy pomocou systému alebo produktu, čo priamo ovplyvňuje jeho celkovú spokojnosť a angažovanosť.
4. **Faktor odmeny (Reward, RW):** Tento faktor sa zameriava na pozitívne pocity a odmeny, ktoré používateľ získava pri používaní systému alebo produktu, čo môže zvýšiť jeho motiváciu a angažovanosť.

Spojením týchto faktorov vzniká celková angažovanosť používateľa [25].

2.5.4 Výber vhodnej verzie merania angažovanosti

Skrátená forma dotazníka UES-SF sa odporúča v prípadoch, keď účastníci vykonávajú viaceré úlohy alebo porovnávajú viacero aplikácií [25]. Pre účely testova-

nia virtuálnej reality sme sa rozhodli použiť UES-SF. Autori práce [25] tiež porovnali UES-LF a UES-SF s inými existujúcimi metódami merania UX, ako napríklad základným dotazníkom SUS a NASA-TLX. Z výsledkov vyplýva, že dotazníky UES a UES-SF poskytujú presnejšie meranie používateľskej angažovanosti vo virtuálnej realite v porovnaní s inými metódami, čo posilňuje dôveru v ich využitie [25].

2.5.5 Spôsob hodnotenia UES

Skóre by malo byť vypočítané ako priemer pre každú podškálu (faktor). Otázky možno jemne prispôbiť kontextu problému a merania, avšak ich počet a význam by sa nemali meniť. Pre výpočet celkového skóre angažovanosti sa má sčítať priemer zo všetkých štyroch podškál verzie UES-LF. V prípade skrátenej verzie (UES-SF) sa vypočíta súčet všetkých odpovedí a následne sa delí dvanástimi, keďže každá zo štyroch podškál obsahuje tri položky. Váhovanie jednotlivých otázok nie je potrebné. Pri meraní vo viacerých iteráciách sa výsledné skóre počíta ako priemer každej iterácie, čím sa získa hodnota v rozmedzí od 1 do 5, kde 5 značí najlepšie hodnotenie a 1 najhoršie [25].

3 Analytická časť - NDMVR

V tejto kapitole sa zameriavame na analýzu softvérového komponentu s názvom NDMVR (N-DiMensional Virtual Reality). Nadväzujúc na predchádzajúcu analýzu uvedenú v práci [7], predstavíme základné charakteristiky komponentu, zhodnotíme jeho aktuálny stav a identifikujeme realizované zmeny. Uvedená práca [7] sa primárne zameriavala na zlepšenie používateľského zážitku pri interakcii s týmto systémom. Cieľom tejto práce je nadviazať na získané poznatky a ďalej ich rozvíjať v kontexte aktuálneho vývoja používateľského rozhrania, so zreteľom na jeho optimalizáciu, udržanie konzistentného používateľského zážitku (UX) a potenciálne zavedenie nových prístupov, ktoré by prispeli k jeho zefektívneniu.

3.1 Možnosti a využitie komponentu NDMVR

Vedecké experimenty zamerané na urýchľovače častíc generujú rozsiahle množstvo komplexných dát. Tieto údaje sú následne analyzované odborníkmi, ktorí na ich základe formulujú vedecké závery a hypotézy. Na podporu tohto procesu bol vyvinutý komponent NDMVR¹, ktorý umožňuje vizualizáciu dvoj- a trojrozmerných histogramov (TH2 a TH3) v prostredí webovej virtuálnej reality, a tým zefektívňuje analýzu experimentálnych údajov [6].

Vizualizácia údajov je realizovaná prostredníctvom histogramov, ktoré patria medzi najstaršie a zároveň najpoužívanější grafické nástroje na reprezentáciu dátových rozložení. Kľúčovým parametrom pri tvorbe histogramu je šírka binu (nazývaná aj šírka triedy alebo intervalu), ktorá určuje dĺžku podintervalov na číselnej osi, tzv. binov, na ktorých je histogram založený [26]. Tento spôsob reprezentácie údajov zaviedol Karl Pearson, ktorý je považovaný za zakladateľa moderného štatistického histogramu [27].

Komponent NDMVR je navrhnutý ako modulárne a opakovane použiteľné riešenie, postavené na softvérových rámcoch A-Frame² a React.js [8]. Ako bolo

¹<https://ndmspc.gitlab.io/ndmvr>

²<https://aframe.io/docs/1.7.0/introduction/>

vysvetlené v kapitole 1, A-Frame zabezpečuje 3D vizualizáciu a podporu pre rôzne hardvérové zariadenia z oblasti virtuálnej a rozšírenej reality, zatiaľ čo React.js optimalizuje výkonnosť aplikácie a umožňuje komponentu vysokú flexibilitu a znovupoužiteľnosť [6].

Podľa autora systému spočíva hlavný prínos NDMVR v porovnaní s tradičnými metódami analýzy údajov v jeho schopnosti sprostredkovať komplexný a priestorovo ucelený pohľad na dáta. Tento globálny pohľad umožňuje výskumníkom efektívnejšie identifikovať štruktúry, vzorce a odchýlky, ktoré by pri klasickom, dvojrozmernom alebo tabuľkovom spracovaní mohli zostať nepovšimnuté. Práve táto vizualizačná dimenzia predstavuje zásadný kvalitatívny rozdiel pri formulovaní hypotéz a interpretácii výsledkov v rámci vedeckého skúmania.

Táto časť práce nadväzuje na predchádzajúci výskum v oblasti vizualizácie experimentálnych údajov v prostredí rozšírenej reality a zameriava sa na zlepšenie používateľského rozhrania a celkového používateľského zážitku [7]. Základy tohto výskumu položil Ing. Martin Fekete [28], a následne ich rozvinul Bc. Lukáš Mikula, ktorý popísal a zdokonalil možnosti vizualizácie vedeckých dát vo webovom prostredí rozšírenej reality [29].

Interakcia s údajmi je možná prostredníctvom bežne dostupných zariadení, ako sú monitor, myš a klávesnica, ako aj pomocou stereoskopických zariadení s vlastnými ovládačmi, ktoré poskytujú používateľovi bohatší a realistickejší zážitok.

3.2 Aktuálny stav riešenia

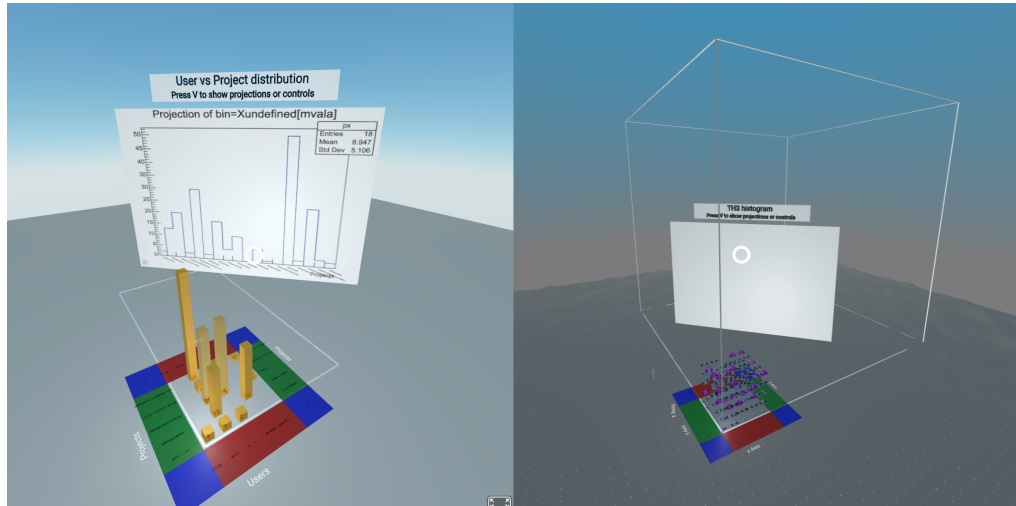
Aktuálna verzia komponentu NDMVR je verejne dostupná online³. Pri analýze jeho architektúry možno identifikovať viacero základných vrstiev. Kľúčovým prvkom je samotný NDMVR balík, ktorý je možné integrovať do takmer akejkolvek webovej aplikácie vytvorenej v rámci technológie React.js.

Pod touto vrstvou sa nachádza VR scéna, ktorá je zodpovedná za samotné vykresľovanie trojrozmerného prostredia. Táto scéna vizualizuje dvojrozmerné a trojrozmerné histogramy typu TH2 a TH3. Proces vizualizácie je zabezpečený prostredníctvom komponentov ako sú VR kamera, 3D histogram a okolitá scéna. Histogram komponent má na starosti samotné vykreslenie dát, zatiaľ čo VR kamera definuje perspektívu pohľadu používateľa (tzv. first-person view). Komponent zodpovedný za zobrazenie okolitého prostredia načíta východiskovú scénu podľa preddefinovaných schém a konfigurácií, ktorú je možné následne modifi-

³<https://ndmspc.gitlab.io/ndmvr>

kovať prostredníctvom rozhrania na tablete.

Vizualizované údaje sú importované vo formáte ROOT Data Analysis Framework. Na ich spracovanie a vykreslenie využíva NDMVR knižnicu JSROOT2, ktorá predstavuje existujúci rámec určený na webovú vizualizáciu a analýzu vedeckých údajov. Táto knižnica sa zároveň používa aj pri vykresľovaní jednoduchých histogramov typu TH1, najmä pri zobrazení jednotlivých binov v trojrozmernej scéne.



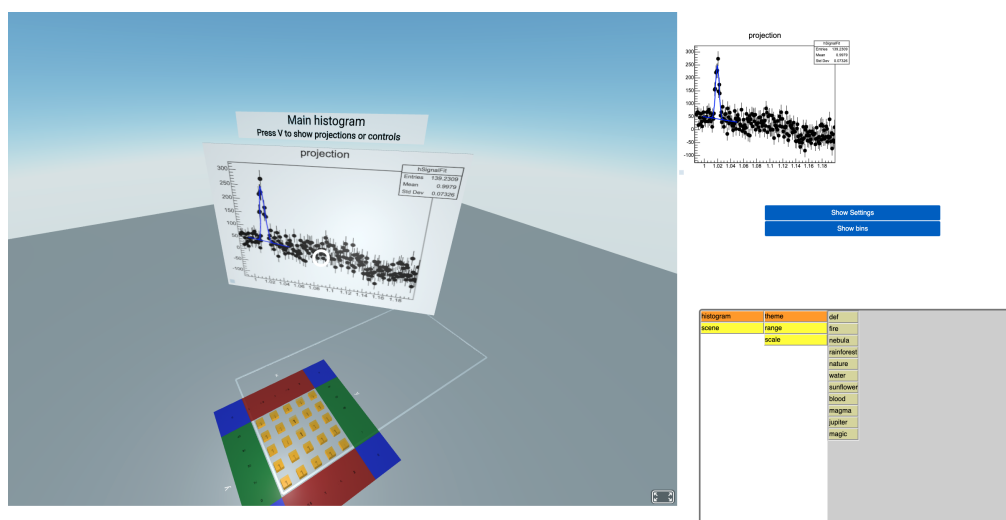
Obr. 3.1: Príklad vizualizácie histogramov typu TH3 a TH2 v NDMVR (zdroj: NDMVR)

3.2.1 Ovládacie prvky riešenia

Po načítaní komponentu a vykreslení scény má používateľ k dispozícii pomocný banner s inštrukciou na aktiváciu ovládacieho tabletu. Prostredníctvom tohto tabletu má prístup k interaktívnym prvkom, ktoré umožňujú hlbšiu a detailnejšiu prispôbenú analýzu vizualizovaných dát. Medzi dostupné možnosti patrí:

- výber VR scény
- úprava mierky zobrazenia binov
- nastavenie rozsahu zobrazovaných binov
- zmena témy binov

VR scéna obsahuje aj pomocné prvky – tenké línie, ktoré reprezentujú celkovú štruktúru histogramu. Používateľ môže prechádzať histogram postupne po segmentoch, čo napomáha zníženiu záťaže grafického procesora (GPU). Rozmery



Obr. 3.2: Desktop zobrazenie NDMVR (zdroj: NDMVR)

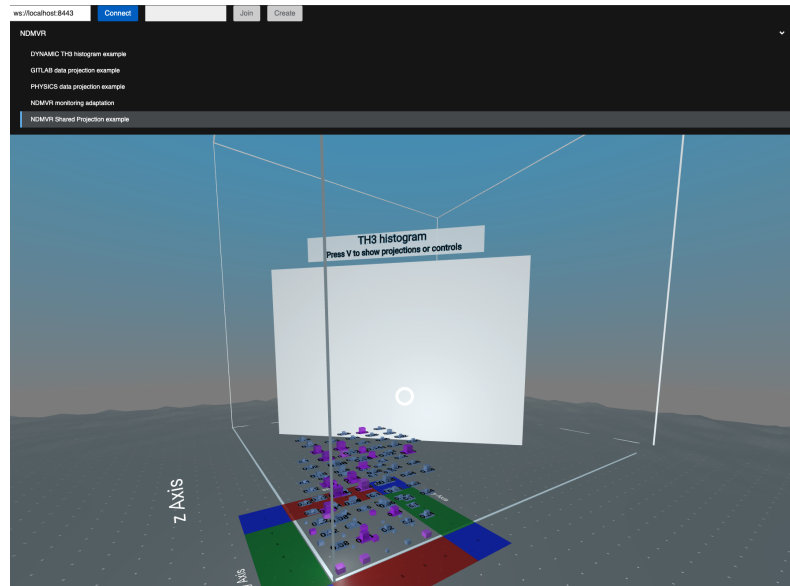
aktívne analyzovanej časti histogramu je možné flexibilne upravovať pomocou ovládacích prvkov dostupných na tablete.

Tablet poskytuje taktiež prístup k ďalším funkciám uvedeným v práci [7], ako je zmena mierky zobrazenia, nastavenie farieb binov a ďalšie. Tieto ovládacie prvky sú dostupné vo verzii pre desktop aj pre VR zariadenia.

Dôležitou súčasťou používateľského rozhrania je horná ovládacia lišta, ktorá umožňuje výber z viacerých demonštračných scenárov využitia komponentu NDMVR. V súčasnosti sú k dispozícii ukážky vizualizácie histogramov typu TH3 a TH2 (pozri obr. 3.1), ktoré reprezentujú fyzikálne údaje, ako aj zobrazenie dát importovaných z platformy GitLab. Cieľom týchto príkladov je ilustrovať univerzálne možnosti využitia komponentu nielen v oblasti fyziky a vedeckej analýzy dát. Súčasťou dema je aj ukážka z práce [29], v ktorej bola implementovaná kolaboračná VR interakcia (pozri obr. 3.3).

V rámci tejto práce bola navyše vytvorená aj nová, odľahčená verzia komponentu, určená výlučne na UX testovanie. Táto verzia kladie dôraz na jednoduchšie, intuitívne používanie a slúži ako pomocný nástroj pri prezentácii cieľov systému respondentom v rámci testovania používateľského zážitku opísaného v kapitole 5.

Komponent NDMVR ďalej ponúka bočné ovládacie prvky v pravej časti desktopového rozhrania. Tieto prvky umožňujú zobrazenie projekcie binov a poskytujú možnosť konfigurácie vizualizačných parametrov, ako sú dátové zdroje, projekčná rovina (napr. *pt*, *multiplicita*, alebo ich kombinácie). Samostatné tlačidlo s označením „bins“ umožňuje priamu úpravu nastavenia projekčných binov. Základné ovládacie prvky desktopového zobrazenia komponentu NDMVR sú zná-



Obr. 3.3: Demo projekcia: Kolaboratívna VR (zdroj: NDMVR)

zornené na obrázku 3.2.

Na záver, v ľavom dolnom rohu obrazovky sa nachádza tlačidlo, ktoré je súčasťou frameworku A-Frame a slúži na vstup do plného VR režimu. Aktiváciou tohto režimu sa deaktivuje okolitý webový obsah a používateľ je presunutý do samostatného, imerzívneho 3D prostredia.

3.3 Stav používateľského rozhrania a návrh ďalších úprav

Z pohľadu používateľského rozhrania (UI) prešiel komponent NDMVR od čias práce [7] určitými optimalizáciami. Na vývoji systému sa podieľali študenti, pričom v niektorých prípadoch došlo k výraznejšej reimplementácii komponentov, čo prinieslo pozitívne, ale aj problematické zmeny.

Jedným z najzásadnejších zásahov bola opätovná implementácia ovládacieho tabletu. Ten aktuálne ponúka čistejšiu štruktúru a jednotnejší render naprieč desktopovou a VR verziou. Z funkčného hľadiska však narážame na viaceré nedostatky. Po interakcii s binom sa jeho funkcionality zastaví, čím dochádza k strate možnosti ďalšej analýzy. Vizualizácia a implementácia tabletu sa navyše líši medzi oboma režimami: vo VR zobrazení predstavuje jediný interakčný prostriedok, pričom jeho zobrazovacia plocha je umiestnená na ruke používateľa. Táto poloha môže byť za istých okolností rušivá, pretože zasahuje do zorného poľa, a to najmä v prípadoch, keď je aktívna selekcia so širokým spektrom funkcií.

V desktopovej verzii je k dispozícii možnosť doplniť tablet o ďalšie ovládacie prvky v podobe HTML komponenty. Tieto elementy však nie sú po vstupe do VR režimu zobraziteľné, pretože nie sú súčasťou samotnej VR scény vytvárajanej pomocou frameworku A-Frame. Tento nedostatok v rámci UI nás motivuje navrhnuť nový, interaktívnejší prístup, ktorý bude využiteľný nielen pri plánovanom UX testovaní, ale aj ako základ pre rozšíriteľné používateľské rozhranie vhodné priamo pre VR prostredie.

Cieľom tohto návrhu je vytvoriť používateľsky komfortné používateľské rozhranie (UI) priamo v prostredí A-Frame, ktoré umožní testovanie bez potreby snímania headsetu. Tým sa eliminuje skreslenie výsledkov spôsobené subjektívnymi dojmami po odložení zariadenia, ako je opísané v časti 2.3.2. Tento aspekt bude pri návrhu plne zohľadnený.

Okrem uvedeného bola reimplementovaná aj pohybová funkcionálna, pôvodne navrhnutá v práci [7]. Nová verzia túto funkcionálnu narušila, čo má negatívny dopad na VR zobrazenie. Tento problém bude podrobne analyzovaný a odstránený v časti 4.1.

Pri testovaní Fly režimu aktuálnej verzie systému bol navyše pozorovaný mierne trhavý pohyb kamery a nepravidelné aktualizovanie jej polohy. Tento technický nedostatok sa pokúsime bližšie objasniť a následne odstrániť v časti 4.1.2.

Ďalšie návrhy zmien sa budú odvíjať od získaných výstupov testovania. Toto testovanie bude reflektovať a implementovať poskytovanie základných nevyhnutných častí UI. Prvou implementačnou prioritou bude analýza a následná implementácia rozšíriteľného a znovu použiteľného zobrazenia UI prostredníctvom komponentu vrchnej úrovne zobrazenia, ktorý bude flexibilne prispôsobiteľný pre rôzne konfiguračné účely systému.

4 Príprava prostredia NDMVR na UX testovanie

Kapitola sa zameriava na odstránenie nedostatkov komponentu NDMVR, ktoré boli identifikované v analytickej časti aktuálneho stavu systému 3.3. Tieto nedostatky budú riešené prostredníctvom návrhu a následnej implementácie požadovaných špecifikácií do systému.

4.1 Pohyb v prostredí NDMVR

Táto časť sa zameriava na odstránenie nedostatkov komponentu určeného na voľný, smerovo orientovaný pohyb v prostredí NDMVR. Rieši úpravu a vylepšenie VR komponentu vytvoreného v rámci práce [7], ktorý slúži na navigáciu v trojrozmernom priestore. Zároveň sa analyzuje jeho rozšírenie a úprava vybraných funkcionalít pohybu pre potreby desktopového zobrazenia.

4.1.1 Opis nedostatkov a návrh zmien

Problémy a návrh zlepšení pre VR zobrazenie

Komponent používaný v rámci VR zobrazenia mal nesprávne orientovaný smer ovládania pohybu pomocou thumbstick kontroléra. Po analýze zmien v rámci verzie NDMVR, uvedených v časti 3.3, bolo zistené, že tieto nedostatky vznikli prehodením orientácie osí, ktoré boli nesprávne inicializované už v prvej verzii projektu. Tento problém je možné ľahko odstrániť, pretože použitý komponent v rámci práce [7] bol postavený na základnej schéme komponentu určeného pre pohyb v priestore, ktorý obsahoval možnosť reverzácie osí, respektíve zmenu predvolenej inicializácie premenných pre reverznú rotáciu. Tento krok by mal daný problém odstrániť.

Pri analýze tohto problému boli odhalené aj ďalšie nedostatky v režime lietačieho módu, ktorý odráža rovnakú orientáciu. Okrem toho možnosť automatic-

kého pristátia nesledovala minimálnu povolenú výšku postavy (teda kamery) a umožňovala opätovné zníženie, čo negatívne ovplyvňovalo používateľský zážitok, najmä pri neúmyselnom viacnásobnom stlačení.

V aktuálnej implementácii NDMVR bolo spozorované neobmedzené časové okno pre dvojité stlačenie triggeru pri pristátí, čo spôsobovalo nežiaduce aktivácie. Pri aktivácii letového režimu jeho držaním dochádzalo k nechcenému pristátiu. Nedostatky spojené s lietaním a pristávaním by sa dali odstrániť úpravou citlivosti rozhrania komponentu.

Problémy a návrhy pre desktopové zobrazenie

Pri desktopovom zobrazení bol doteraz používaný defaultný komponent pre pohyb kamery uvedený v dokumentácii A-Frame, označený ako "WASD controls"¹. Tento komponent zabezpečoval pohyb v priestore v závislosti od smeru rotácie kamery. Doterajšie použitie umožňovalo pohyb dopredu, dozadu, doľava a doprava. Pre účely lepšej analýzy boli v práci [7] pridané možnosti pohybu nahor a nadol po osi Y.

Tieto akcie boli vykonávané na strane Reactu a nie priamo na čistom A-Frame komponente. Posun sa realizoval len ako nutná akcia, ktorá presne definovala jednotku pohybu nahor a nadol, bez akejkoľvek odozvy na stlačenie klávesy, prípadne s jemným efektom pokračovania akcie aj po uvoľnení tlačidla. To viedlo k dojmu sekaného pohybu. Aj napriek použitej interpolácii medzi dvomi bodmi, uvedenej v práci [7], bol pohyb nedostatočný. Preto sme sa rozhodli, že na dosiahnutie požadovaného zlepšenia citlivosti a plynulosti pohybu, ako aj dosiahnutia podobného správania ako pri pohybe po osiach X a Z, sa tento problém bude riešiť výlučne na strane A-Frame komponentu.

Pre efektívnejšie používanie možnosti zmeny výšky kamery, teda postavy, sme sa rozhodli pridať možnosť automatického pristátia, podobne ako v komponente pre VR zobrazenie, ako aj možnosť režimu voľného lietania v smere natočenia kamery. Taktiež sme špeciálne pre desktop pridalí funkciu vzlietnutia, kde sa namiesto držania klávesy postava automaticky plynulým pohybom dostane do preddefinovanej výšky pri stlačení klávesy.

Rozhodli sme sa realizovať navrhnuté zmeny v rámci nášho vlastného "**custom**"WASD ovládača pre pohyb kamery. Tento komponent obohatíme o uvedené akcie a zároveň zachováme jeho doterajšie fungovanie, ako aj spôsob realizácie pohybu. Tento prístup nám poskytuje najlepšie používateľské skúsenosti, ako aj najlepšie výsledky podľa skúseností s komponentom NDMVR a spätnej väzby od

¹<https://github.com/aframevr/aframe/tree/master/src/components>

používateľov. Princíp pohybu bol zachovaný podľa najnovšej verzie kódu uvedeneho v knižnici A-Frame 1.5.0 ².

4.1.2 Implementácia a vylepšenie ovládačov pre pohyb kamery

Implementácia zmien pre VR zobrazenie

Opačne orientovaný pohyb v rámci VR zobrazenia bol odstránený zmenou inicializácie premennej na hodnotu True v základnej schéme pre reverznú orientáciu. Do schémy bola pridaná definovaná výška postavy (kamery). V prípade, že výška nie je definovaná, je predvolene nastavená na hodnotu 2 na osi y. Režim pristátia nie je možné aktivovať, ak kamera dosiahne túto predvolenú hodnotu alebo akúkoľvek inú výšku definovanú v schéme. Akcia pristátia je kontrolovaná pred jej vykonaním, a to zachytením príslušného eventu pre pristátie. Pohyb počas animácie pristátia je kontrolovaný so stanovenou inicializačnou hodnotou, čím sa zabráni posunu kamery na nižšie definovanú výšku.

Interval stlačenia dvojitého kliknutia bol obohatený o dolný limit 30 ms, ktorý zabráni náhodnému, nechcenému dvojitému stlačeniu triggeru, ak je tlačidlo držané a jeho intenzita je povolená a následne opäť zmenená. Horná hranica intervalu bola upravená na nižšiu hodnotu, konkrétne na 300 ms, čím sa určuje rýchlosť dvojklíku. Tieto hodnoty boli nastavené na základe testovania komponentu a hľadania optimálneho nastavenia pre príjemný používateľský zážitok.

Vlastný pohybový controller pre desktopovú verziu NDMVR

Implementácia custom WASD kontroléra bola obohatená o nové akcie stlačenia kláves Q a E pre pohyb nahor a nadol v smere osi y. Tento pohyb je rovnaký ako pri stlačení kláves W, A, S, D, ale bol doplnený o vlastný parameter rýchlosti, nakoľko pohyb stúpania a klesania môže byť potrebné vykonávať citlivejšie. Predvolene sú parametre pohybu do strán aj do výšky nastavené na rovnakú rýchlosť.

Pri stlačení klávesy SHIFT je aktivovaný režim voľného lietania, ktorý bol predpripravený v komponente WASD controls. Jeho vlastné použitie a následné prispôbenie do nášho projektu nám umožnilo aktivovať ho iba v prípade stlačenia klávesy SHIFT, ktorú sme považovali za najvhodnejšiu pre neustále držanie, pričom zároveň umožňuje pohyb pomocou kláves WASD. V kombinácii s klávesami Q a E sa vykoná akcia automatického pristátia pri stlačení klávesy Q a plynulý pohyb na predvolenú výšku pri stlačení klávesy E, pričom túto výšku je možné

²<https://github.com/aframevr/aframe/blob/master/src/components/wasd-controls.js>

definovať samostatne v schéme, spolu s minimálnou výškou voči postave, teda kamere.

Bola zabezpečená podmienka minimálneho klesnutia, ktoré je obmedzené len na predvolenú výšku vzhľadom na definovanú výšku kamery, teda 2, ak kamera nemá definovanú inú výšku. Pri držaní kláves E a Q je pohyb maximálne plynulý, nakoľko nie je definovaný žiadnou fixnou jednotkou, ale smerom a intenzitou držania klávesy, ako aj rýchlosťou operácie. Pohyby sa aktualizujú v prípade zachytenia akcie vykonávania každý frame. Z údajov je vypočítaný nový vektor pohybu, ako aj Eulerova rotácia, a tie sú pridané prostredníctvom funkcie `.add` k aktuálnej polohe kamery.

Pre intenzívnejší používateľský zážitok bol do procesu pristátia pridaný efekt otrasu kamery. Pri pristátí bol zámerne definovaný menší rozdiel (-0.6) pri posune nadol, ak sa dosiahne minimálna hodnota výšky. Tento menší skok simuluje jemné pristátie s malým odrazom od zeme a návrat na definovanú minimálnu pozíciu vzhľadom na pozíciu kamery.

Aktualizácia polohy a rýchlosti

Aktualizácia polohy a rýchlosti kamery je kľúčovou funkcionalitou custom WASD kontroléra, ktorá zabezpečuje plynulý a presný pohyb vo virtuálnom priestore. Tento proces sa realizuje prostredníctvom niekoľkých kľúčových metód a algoritmov, ktoré zohľadňujú vstupy od používateľa, fyzikálne parametre a aktuálny stav systému.

Metóda `tick` je volaná v každom frame a jej hlavnou úlohou je získať aktuálne vstupy od používateľa a následne aktualizovať pozíciu kamery. Táto metóda vykonáva nasledovné kroky:

1. **Kontrola vstupov a rýchlosti:** Najprv sa kontroluje, či existujú aktívne vstupy od používateľa (stlačené klávesy) alebo aktuálna rýchlosť pohybu na niektorej z osí (`adAxis`, `wsAxis`, `qeAxis`). Ak žiadne takéto vstupy neexistujú, metóda končí, čím šetrí výpočtové zdroje.
2. **Normalizácia času:** Časový rozdiel (Δt) je prepočítaný na sekundy, čo umožňuje správnu škálovateľnosť pohybu nezávisle od rýchlosti vykresľovania (frame rate).
3. **Aktualizácia rýchlosti:** Volaním metódy `updateVelocity` sa upravuje aktuálna rýchlosť pohybu kamery na základe aktuálnych vstupov od používateľa a nastavených parametrov akcelerácie.

4. **Aktualizácia pozície:** Ak po aktualizácii rýchlosti existuje nenulová rýchlosť na niektorej z osí, vypočíta sa pohybový vektor a následne sa pripočíta k aktuálnej pozícii kamery pomocou metódy `el.object3D.position.add`.

Metóda `updateVelocity` Metóda `updateVelocity` je zodpovedná za výpočet novej rýchlosti kamery na základe vstupov od používateľa a fyzikálnych parametrov, pričom zohľadňuje nasledujúce faktory:

- **Resetovanie rýchlosti:** Ak časový rozdiel medzi jednotlivými frame-ami prekročí maximálnu povolenú hodnotu (`MAX_DELTA`), rýchlosti na všetkých osiach sú resetované na nulu, aby sa zabránilo nekonzistentnému správaniu.
- **Uvoľnenie rýchlosti:** Použitím faktora uvoľnenia (`scaledEasing`) sa postupne znižuje rýchlosť na každej osi, čím sa simuluje prirodzené spomaľovanie pohybu. Tento faktor je vypočítaný ako mocnina inverzie uvoľnenia k delta času vynásobenému 60.
- **Akcelerácia:** Na základe aktuálne stlačených kláves a nastavených parametrov akcelerácie (`acceleration`, `accelerationY`) sa aktualizuje rýchlosť na príslušných osiach. Napríklad, ak je stlačená klávesa `KeyW` alebo `ArrowUp`, rýchlosť na osi `z` (`wsAxis`) sa zvyšuje, ak je stlačená klávesa `KeyQ`, rýchlosť na osi `y` (`qeAxis`) sa znižuje, ak sa pozícia nachádza nad predvolenou výškou kamery.

Metóda `getMovementVector` vypočíta výsledný pohybový vektor, ktorý je použitý na aktualizáciu pozície kamery v každom frame. Tento vektor zohľadňuje:

- **Smere a veľkosť rýchlosti:** Pohybový vektor je najprv inicializovaný aktuálnou rýchlosťou a následne škálovaný podľa času Δt , aby sa zabezpečil konzistentný pohyb nezávisle od rýchlosti vykresľovania.
- **Rotácia kamery:** Vektor je transformovaný Eulerovou rotáciou kamery, čo umožňuje pohyb v smere, ktorým je kamera otočená. Tento krok je kľúčový pre zabezpečenie intuitívneho ovládania pohybu, kde napríklad pohyb dopredu vždy znamená pohyb v smere pohľadu kamery.

Príklad aktualizácie pozície

Nasledujúci príklad demonštruje, ako je aktualizovaná pozícia kamery v reálnom čase:

```
let velocity = new THREE.Vector3(0, 0, 0);
let rotation = this.el.getAttribute('rotation');
let directionVector = new THREE.Vector3(0, 0, 0);
let rotationEuler = new THREE.Euler(0, 0, 0, 'YXZ');

velocity.x += this.keys.KeyA ? -this.data.acceleration : 0;
velocity.x += this.keys.KeyD ? this.data.acceleration : 0;
velocity.y += this.keys.KeyQ ? -this.data.accelerationY : 0;
velocity.y += this.keys.KeyE ? this.data.accelerationY : 0;
velocity.z += this.keys.KeyW ? -this.data.acceleration : 0;
velocity.z += this.keys.KeyS ? this.data.acceleration : 0;

directionVector.copy(velocity).multiplyScalar(delta);
rotationEuler.set(
    THREE.MathUtils.degToRad(rotation.x),
    THREE.MathUtils.degToRad(rotation.y), 0
);
directionVector.applyEuler(rotationEuler);

this.el.object3D.position.add(directionVector);
```

Vysvetlenie Delt

Delta (Δt) je časový rozdiel medzi dvoma snímkami. Používa sa na normalizáciu pohybu, aby bol pohyb kamery hladký a konzistentný nezávisle od frekvencie vykresľovania (frame rate).

- **velocity** je vektor rýchlosti kamery.
- **rotation** je rotácia kamery.
- **directionVector** je vektor, ktorý určuje smer pohybu.
- **rotationEuler** je Eulerova rotácia, ktorá sa používa na aplikovanie rotácie na pohybový vektor.

Rýchlosť kamery sa aktualizuje na základe stlačených kláves. Pohybový vektor (**directionVector**) sa škáluje pomocou delty a aplikuje sa na neho rotácia kamery. Tento prepočet zabezpečuje, že pohyb kamery je časovo nezávislý a plynulý.

Celkovo tento prístup zabezpečuje, že pohyb kamery je plynulý a intuitívny, pričom umožňuje precízne ovládanie intenzity a smeru pohybu na základe vstupov od používateľa.

4.2 UI menu v prostredí VR

V tejto časti bude vytvorený návrhový a implementačný základ pre používateľské rozhranie (UI) vo forme knižnice, ako aj hierarchicky a komponentovo orientované UI zobrazenie. Toto rozhranie bude konfigurovateľné podľa rôznych požiadaviek zobrazenia, pričom jednou z hlavných aktuálnych potrieb je vytvorenie základného používateľského rozhrania pre UX testovanie, podrobnejšie opísané v kapitole 5.

4.2.1 Opis nedostatkov poskytovaného UI a návrh zmien

Táto časť vychádza z analýzy nedostatkov uvedených v časti 3.3. Uvedené zmeny v reimplementácii tabletu priniesli nedostatky v dizajne, ako aj v podpore rozširiteľnosti. Taktiež neposkytujú dostatočné možnosti na sledovanie priebehu testovania a zabezpečenie zberu dát priamo v prostredí NDMVR, na základe odporúčaní uvedených v časti 2.3.2. Aktuálne prvky UI sú vytvorené priamo v React komponente zo základných primitív A-Frame, ktoré nie sú dostatočne detailne a esteticky spracované, keďže ide len o jednoduché materiály určitej farby. Ich opätovná použiteľnosť je teda minimálna a v prípade pridania ďalších možností je nevyhnutná opätovná implementácia v novej forme.

NDMVR v súčasnosti obsahuje predovšetkým UI orientované na ovládanie tabletu. Pre uvedené nedostatky sme museli zvážiť implementáciu nového prvku UI. Tento nedostatok sme sa rozhodli odstrániť zavedením UI okna, ktoré je v projekte nazývané menu. Toto okno by mohlo priniesť ucelený pohľad na všetky možnosti zobrazenia a interakcie v rámci scény, ale zároveň by fungovalo ako oddelená časť, akýsi displej počítačovej jednotky, ktorý NDMVR poskytuje používateľovi.

Tento UI prvok bude poskytovať chýbajúce možnosti výberu a interakcie na základe rôznych konfigurácií, ako aj podporu pre realizáciu UX experimentov prostredníctvom dotazníkov alebo výkonových testov priamo v prostredí, bez nutnosti zloženia headsetu, čím sa zabráni strate pôvodného a celkového zážitku. Tento krok by mal viesť k zlepšeniu kvality samotných výsledkov testovania, ako aj k rýchlejšiemu a plynulejšiemu priebehu experimentov. Experiment by mal prebiehať úplne a nepretržite v prostredí NDMVR. Zároveň tento krok rieši ne-

dostatky meraní a potenciálne skreslenie výsledkov testov vykonávaných v práci [7]. Pri implementácii bude sledovaná aj možnosť automatizácie a teda rýchleho prístupu k testovaniu po akejkoľvek zmene, pričom sa budú získavať dáta, ktoré budú podliehať analýze.

4.2.2 Prístup k návrhu používateľského rozhrania v prostredí A-frame

Prvým krokom pri návrhu a analýze riešenia bolo preskúmanie možností podpory používateľského rozhrania (UI) v rámci frameworku A-Frame [30], konkrétne vo forme 2D zobrazenia. Cieľom bolo čo najviac sa priblížiť klasickému webovému menu – teda oknu s ponukou možností a tlačidiel, ktoré je známe z bežných webových aplikácií.

Základnou požiadavkou na návrh UI menu bola čo najvernejšia reprodukcia takéhoto komponentu vrátane jeho jednotlivých častí. Framework A-Frame však neposkytuje natívnu podporu pre prácu s komplexným používateľským rozhraním. Príklady uvedené v oficiálnej dokumentácii síce umožňovali vytváranie jednoduchých UI prvkov, no vyžadovali pomerne rozsiahly a neprehľadný zápis kódu, ktorý môže byť komplikovaný najmä pre menej skúsených vývojárov pracujúcich s týmto nástrojom. Navyše, tvorba takéhoto rozhrania bola značne statická a postrádala flexibilitu či dynamickosť, čo je v prípade interaktívneho prostredia VR podstatný nedostatok.

Z tohto dôvodu sme sa rozhodli preskúmať existujúce knižnice tretích strán. Ako vhodné riešenie sa ukázala knižnica A-Frame GUI³, ktorá nám poskytla inšpiráciu pre ďalší postup. Táto knižnica umožňuje registráciu vlastných A-Frame primitív spolu s príslušnými udalosťami (eventmi), pričom tieto udalosti sa môžu zachytávať priamo v komponente už pri jeho vytvorení ako atribúty entity. Takýto prístup poskytuje výrazne lepšie možnosti rozšíriteľnosti, opätovnej použiteľnosti a modulárnosti návrhu používateľského rozhrania v prostredí NDMVR.

Tento prístup sa ukázal ako veľmi vhodný v kombinácii s technológiou React. Vďaka použitiu syntaxe JSX je možné jednoducho vrátiť entitu používateľského rozhrania spolu s ostatnými komponentmi a vytvoriť tak šablónu dynamického zobrazenia na základe konfiguračného súboru.

S cieľom zabezpečiť lepšiu reprezentáciu a opätovnú použiteľnosť rozhrania bolo potrebné získať vhodný layout zobrazenia. Na tento účel bol využitý balík ⁴, ktorý poskytuje okrem iného aj komponent Window. Tento komponent umožňuje

³<https://github.com/rdub80/aframe-gui>

⁴<https://github.com/binzume/aframe-xylayout>

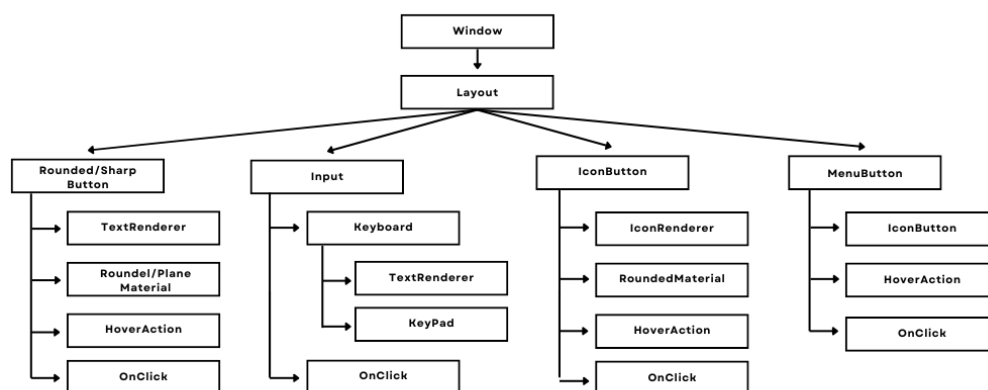
manipuláciu s oknom (napr. jeho presúvanie), zobrazenie rozloženia a zároveň ponúka 2D plochu pre umiestnenie ďalších prvkov používateľského rozhrania.

Hoci balík obsahoval aj ďalšie komponenty, tie však nespĺňali naše požiadavky na dizajn, funkčnosť či mieru opätovného využitia. Z tohto dôvodu sme sa rozhodli vytvoriť ostatné používateľské prvky pre menu nezávisle a samostatne.

Pri návrhu týchto používateľských prvkov sme využili základné vlastnosti frameworku A-Frame, vrátane jeho primitív, možností geometrie, materiálov, ikon, animácií a definovaného atribútu onclick pre každú interaktívnu (klikateľnú) entitu. Názov tohto atribútu bol zvolený zámerne, pretože bežný onClick je rezervovaný a v komunite známy pre React. Pre tento atribút v A-Frame sme zvolili malé písmená, keďže je vykonávaný na strane A-Frame a následne odovzdávaný Reactu.

Základnou myšlienkou bolo, aby každá entita (resp. vnorený A-Frame komponent) dokázala spracovať konkrétnu udalosť – napríklad kliknutie – a vykonala príslušnú funkciu, ktorá jej bola pridelená. Tento atribút je možné opakovane využívať a volať pre akúkoľvek UI entitu definovanú v rámci JSX štruktúry.

Na obrázku 4.1 je zobrazené hierarchické usporiadanie UI menu s použitím postupného renderovania na ploche okna podľa definovaného prispôsobiteľného layoutu a primitív poskytnutých výslednou knižnicou tejto práce, v podobe registrovaných prvkov UI a ich podkomponentov.



Obr. 4.1: Grafický prehľad hierarchickej komponentovej štruktúry riešenia UI Menu primitív v A-Frame

4.2.3 Grafický dizajn a funkčné rozmiestnenie UI menu

UI menu by malo obsahovať komponent na navigáciu a zobrazovacie okno, ktorého obsah sa bude dynamicky obnovovať podľa vybraného komponentu na vy-

kreslenie. Zobrazenie musí byť čitateľné, dostatočne veľké a zabezpečiť jednotný zážitok na headsete, ako aj v desktopovej verzii.

Pre navigáciu medzi možnosťami bude vytvorený komponent, ktorý sa dynamicky generuje na základe súboru UI-menu-konfigurácia JSON. Tento súbor určuje nielen dizajn, ale aj rozloženie navigačnej lišty. Navigačná lišta je umiestnená v spodnej časti obrazovky s miernym sklonom 15 stupňov, čím umožňuje dodatočné ovládanie pomocou hand trackingu a Oculus controllera, čo zlepšuje presnosť klikania.

Štýl a umiestnenie navigačnej lišty sú navrhnuté podľa rozhrania Oculus OS zobrazenom na obr. 4.2, čím sa dosiahlo intuitívne ovládanie a jednoduché pochopenie. Zobrazovacie okno je umiestnené mierne nad komponentom <navbar> a je zarovnané tak, aby bolo dobre viditeľné a funkčné. Inšpirácia rozhraním Oculus menu zabezpečuje intuitívne a známe ovládanie, ako aj rýchle pochopenie zobrazenia.



Obr. 4.2: Oculus UI domovské prostredie (zdroj: [31])

4.2.4 Implementácia navrhnutého UI menu a jeho optimalizácia

Všetky primitíva použité v komponentoch sú uložené ako knižničné súbory projektu v priečinku `lib/aframe/gui`. Ide o modifikáciu pôvodnej knižnice ⁵, ktorá

⁵<https://github.com/binzume/aframe-xylayout>

bola rozšírená o nové vlastné primitíva, ako sú obdĺžnik so zaoblenými rohmi vytvorený pomocou Three.js, zobrazenie ikon v ráme, ikonové tlačidlo, textové tlačidlo, zaoblené tlačidlo, modifikované scrollovacie okno pre obsah a tlačidlo „menu-icon“, vytvorené pre navigačné možnosti a animované reakcie v komponente <navbar>. Súčasťou implementácie sú aj pomocné (utils) komponenty, ktoré zabezpečujú správne zobrazovanie, funkcionálnu jednotlivosť primitív, ako aj chod celého menu.

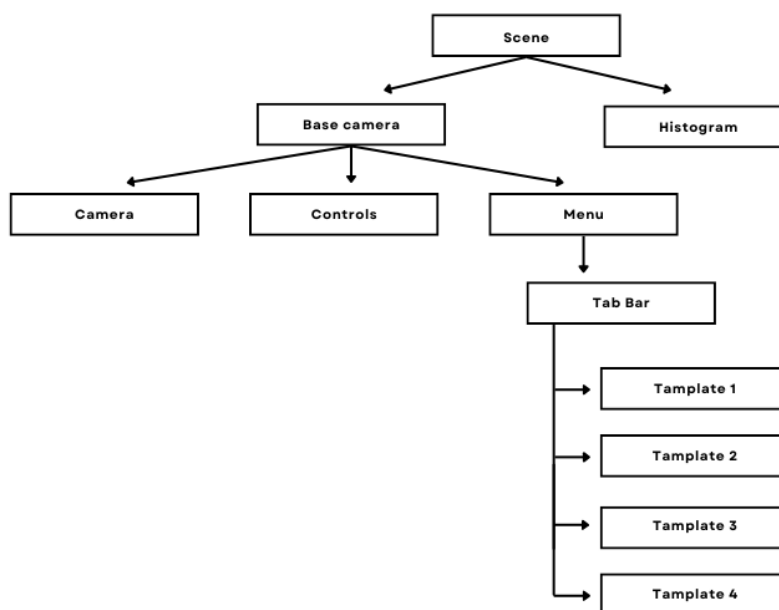
Konštrukciu riadi centralizovaný komponent, ktorý spravuje celé zobrazenie navigačnej lišty (navbar) a hlavného obsahového okna (content window). Komponent <Menu> slúži na načítanie konfiguračného súboru, vykreslenie navigačnej lišty a odovzdávanie vlastností (props) komponentom, ktoré sú v tejto lište definované. Tieto vlastnosti nesú dáta z konfiguračného JSON súboru a sú využité v každom komponente, ktorý je do rozhrania vložený prostredníctvom navigácie. Takýto prístup umožňuje jednoduchú modifikáciu, pridávanie nových prvkov alebo ich úpravu bez zásahu do štruktúry komponentov, a to výhradne cez konfiguračný súbor.

Tieto komponenty sú v rámci projektu označované ako „template“. Každý template má preddefinovaný layout, pričom jeho obsah je dynamicky prispôbovaný na základe údajov špecifikovaných v príslušnej časti JSON objektu. V rámci komponentu <Menu> sa nachádza logika, ktorá každému template priraduje jeho vlastnosti a zabezpečuje vykreslenie relevantného obsahu. Tento spôsob výrazne zjednodušuje tvorbu a správu ďalších komponentov bez potreby manuálneho definovania nových primitív alebo ich usporiadania.

Na obr. 4.3 je znázornená hierarchická štruktúra použitých komponentov, ktoré sa podieľajú na renderovaní UI menu a jeho jednotlivých template komponentov. Samotný proces zobrazovania je realizovaný podľa nasledujúcej logiky: obsah sa vykresľuje v zobrazovacom okne, pričom predchádzajúci obsah je pred načítaním nového odstránený zo scény. V prípade zmeny sa nový obsah nanovo vloží, čo výrazne prispieva k zníženiu záťaže na systém a optimalizácii veľkosti scény.

Táto optimalizácia je obzvlášť dôležitá pri použití Oculus zariadení, ktoré sú výkonnostne obmedzené. Zachovanie čo najmenšieho množstva aktívnych komponentov v scéne znižuje riziko prepadu snímkovej frekvencie a zabezpečuje plynulý používateľský zážitok. Skryté alebo neaktívne prvky, ktoré by ostávali v scéne, by mohli mať negatívny vplyv na výkon aplikácie.

UI menu je možné otvárať a zatvárať prostredníctvom klávesu **X** v desktopovej verzii, čo zabezpečuje A-frame komponent menu-visibility. V prípade použitia na Oculus zariadeniach je na túto akciu priradené tlačidlo **B**, ktorého akciu



Obr. 4.3: Grafický prehľad hierarchickej komponentovej štruktúry riešenia UI Menu v Reacte

vykonáva komponent `b-button-listener`.

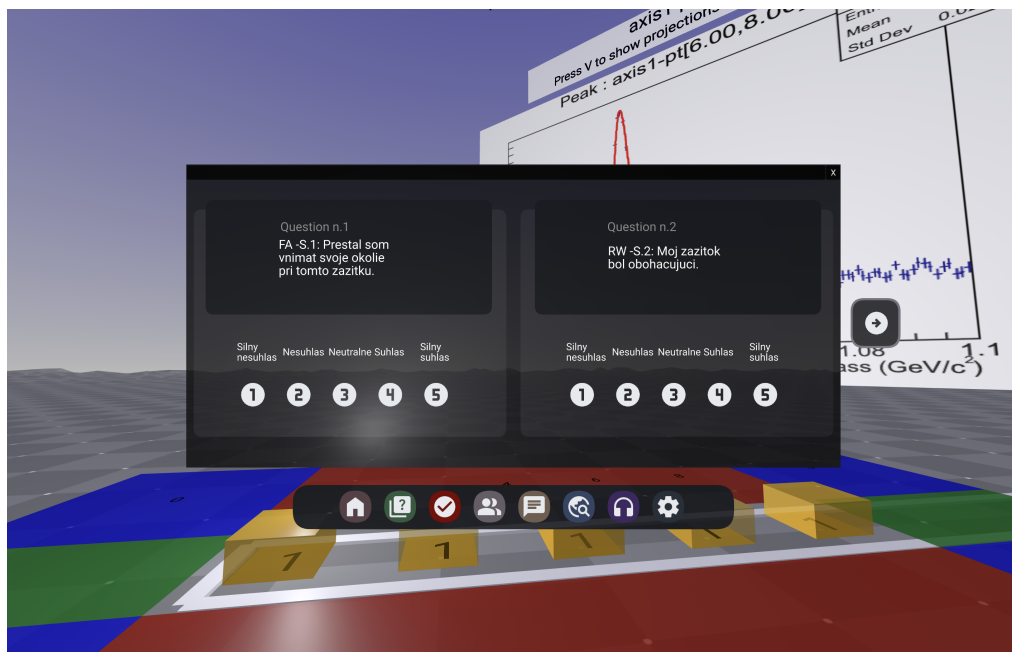
Pre UI menu bol vytvorený komponent, ktorý kalkuluje natočenie a vzdialenosť menu na základe konštantne definovaných vstupných parametrov. Menu je tak používateľovi dostupné vždy v rovnakej vzdialenosti a vždy priamo v smere jeho pohľadu. Tieto funkcie zabezpečujú pomocné komponenty implementované ako súčasť knižnice vytvorenej v rámci tejto práce.

4.2.5 Opis šablón (templates) pre UI menu v NDMVR

V prostredí NDMVR boli vytvorené tri hlavné šablóny pre UI menu, z ktorých každá slúži na odlišný účel:

Interaktívna šablóna s možnosťou výberu odpovedí

Táto šablóna, znázornená na obr. 4.4, je určená na čítanie a posúvanie textu, ako aj na výber (zaškrtnutie) jednotlivých možností. Využíva sa najmä na účely testovania, ako je podrobnejšie opísané v kapitole 5. Umožňuje používateľovi vyjadriť svoj postoj k jednotlivým tvrdeniam vo forme zaznamenania odpovede a jej odoslania prostredníctvom služby *emailJS*, kde je odpoveď uložená a prístupná na ďalšiu analýzu.



Obr. 4.4: Vytvorený template pre Music Player (zdroj: NDMVR)

Šablóna pre sledovanie postupu a hodnotenie krokov

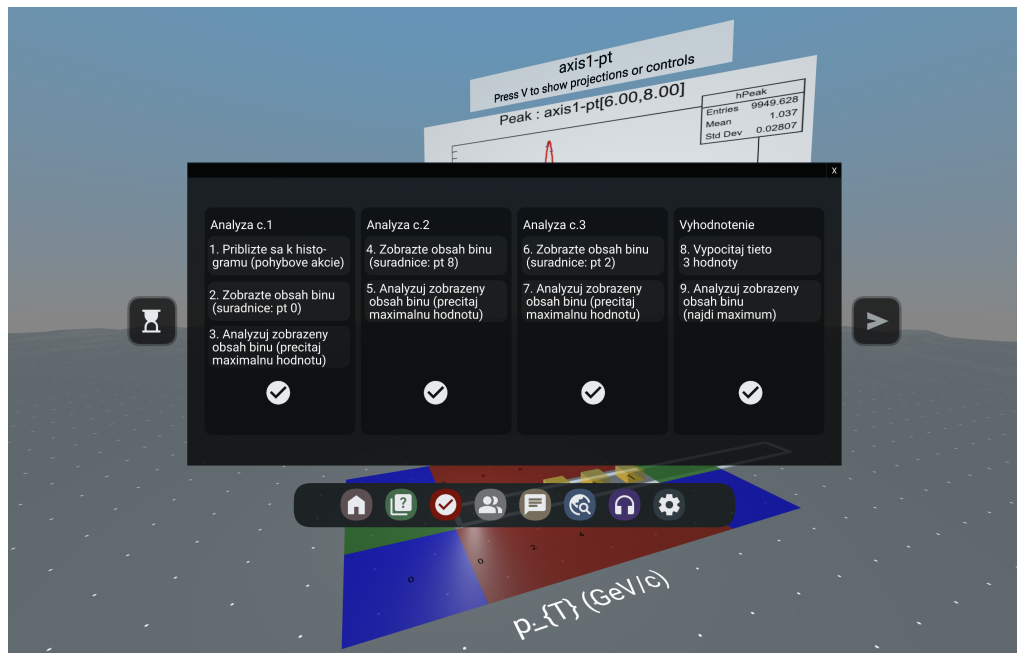
Šablóna uvedená na obr. 4.5 bola navrhnutá pre úlohy, kde používateľ označuje úspešnosť vykonaných krokov alebo úkonov. Využitím sledovania časového intervalu vykonávania úloh a ich následného odoslania na e-mailovú adresu, tento prístup umožňuje realizovať behaviorálne sledovanie používateľa počas testovania, ako je podrobnejšie opísané v kapitole 5. Tento typ zobrazenia poskytuje používateľom prehľad o krokoch, ktoré majú splniť, a zaznamenáva ich pokrok.

Šablóna pre tile zobrazenie a správu prehrávania skladieb

Zobrazenie šablóny uvedenej na obr. 4.6, ktorá je určená na vykreslenie viacerých tile možností v rozložení 3x3, bola implementovaná v rámci hudobného prehrávača v NDMVR. Tento prehrávač umožňuje podľa definovanej konfigurácie skladieb a URL adresy zdroja spustenie danej skladby vo VR prostredí, čím je podporovaná analýza a predovšetkým adaptabilita používateľským požiadavkám.

Rozšírené vyhľadávanie a spracovanie textového vstupu používateľa

Okrem týchto šablón sme vytvorili aj **prehliadač** zobrazený na obr. 4.7, ktorý slúži ako ukázkový komponent pre interakciu s virtuálnou klávesnicou, ktorú UI menu poskytuje na zadávanie vstupných údajov. Tento komponent nie je šablónou, ale pevne definovaným prvkom, pretože poskytuje iba základnú funkcionality na získanie vstupu od používateľa a následné presmerovanie na predvolený prehliadač.



Obr. 4.5: Vytvorený template pre sledovanie úspešnosti krokov alebo úkonov (zdroj: NDMVR)



Obr. 4.6: Vytvorený template pre Music Player (zdroj: NDMVR)

dač zariadenia pre vyhľadávanie informácií. Služi ako ukážka získania textového vstupu a jeho prenosu medzi A-frame a Reactom.



Obr. 4.7: Vytvorený komponent pre webové vyhľadávanie, praktická ukážka spracovania klávesového vstupu od používateľa (zdroj: NDMVR)

4.2.6 Znovupoužiteľnosť a rozšíriteľnosť používateľského rozhrania (UI)

Implementácia pridávania nových zobrazení a komponentov do UI menu v NDMVR je realizovaná nasledovne:

1. **Vytvorenie komponentu a jeho registrácia:** Prvým krokom je vytvorenie vlastného komponentu, ktorý sa následne zaregistruje do `<navbar>`. Tento komponent bude slúžiť ako základ pre nové zobrazenie alebo možnosť v UI menu.
2. **Priradenie template a konfigurácie:** Po vytvorení komponentu sa priradí nový template k `<menu>` komponentu, spolu s jeho príslušnou konfiguráciou. Template definuje vzhľad a správanie nového menu prvku.
3. **Implementácia UI entít:** V rámci vytvoreného komponentu sa implementujú ľubovoľné UI entity z poskytnutej knižnice, pričom sa môžu využiť aj ďalšie základné A-frame entity pre rozšírenie funkcií. Tento krok umožňuje flexibilitu pri definovaní rôznych typov zobrazení.

Pre optimalizáciu a presné umiestnenie objektov sa odporúča používať `a-entity` na obalenie primitív z knižnice. Táto entita slúži ako nadradený kontajner a je obzvlášť užitočná pri definovaní presných pozícií objektov bez nutnosti použiť

layout. Ak je však požiadavka na použitie preddefinovaného layoutu, vhodné je využiť entitu `xy-container`, ktorá ponúka registrovaný layout s rôznymi možnosťami usporiadania objektov. Tento layout je detailne popísaný v dokumentácii⁶.

Tento implementačný prístup umožňuje vytvorenie komponentu, ktorý sa následne zobrazuje v obsahovom okne po jeho aktivácii cez `<navbar>`. Takto definovaný prístup prináša používateľom flexibilitu pri pridávaní nových menu komponentov, čím zabezpečuje jednoduchú a intuitívnu aplikáciu zmien v rozhraní.

⁶<https://binzume.github.io/aframe-xylayout/examples/layout.html>

5 Realizácia a hodnotenie testovania respondentov

Táto kapitola je zameraná na testovanie respondentov. Obsahuje návrh metodiky testovania, samotnú realizáciu testovacieho procesu, zber výsledkov, ich analýzu a následné vyhodnotenie na základe získaných údajov. Súčasťou analýzy je aj zohľadnenie možných nedostatkov navrhovaného riešenia a identifikácia slabých miest z pohľadu strategického plánovania a realizácie testovania. Kapitola zároveň slúži ako praktická ukážka aplikácie vybraných riešení opísaných v časti 4.2 a poskytuje východisko pre kapitolu 6, ktorá sa venuje implementácii týchto zmien.

5.1 Návrh metodiky testovania

Pre samotné testovanie sme sa rozhodli vylepšiť prístupy opísané v práci [7]. Jedným z hlavných cieľov bolo zefektívniť spôsob vykonávania hodnotiacich metrík v realistickom prostredí. Ako vyplýva z analýzy prezentovanej v časti 2.3, kde bola diskutovaná problematika straty dôveryhodnosti výsledkov a narušenia celkového používateľského zážitku, rozhodli sme sa pre modifikáciu testovania tak, aby bolo možné zaznamenávať výstupy priamo prostredníctvom VR headsetu.

Zvažovaná bola aj možnosť využitia externých riešení tretích strán, ako napríklad Google Forms, ktoré je možné spúšťať a vyplňovať priamo vo VR prostredí. Praktické testovanie (realizované autormi a viacerými dobrovoľníkmi mimo hlavnej testovacej skupiny) však preukázalo, že tento prístup nie je optimálny. Spustenie Google Forms si vyžadovalo opustenie režimu NDMVR, otvorenie webového prehliadača a následné vyplnenie formulára, čo výrazne narúšalo plynulosť zážitku a mohlo viesť k skresleniu získaných údajov.

Na základe týchto zistení bolo pristúpené k vývoju a implementácii vlastného riešenia, popísaného v časti 4.2.4, ktoré bolo integrované priamo do prostredia NDMVR. Takto navrhnuté riešenie eliminovalo potrebu prechodu mimo aplikáciu alebo vykonávania nadbytočných interakcií, čím sa výrazne znížilo riziko na-

rušenia používateľského zážitku a zároveň sa zvýšila spoľahlivosť zozbieraných dát.

Kľúčové vlastnosti navrhovanej metodiky:

- **Dodržanie odporúčaní pre testovanie UX vo VR**
- **Plynulý prechod na hodnotenie**
- **Minimalizácia narušenia zážitku**
- **Samostatnosť v interakciách**

Navrhovaný prístup umožňuje efektívne vykonávať testovanie bez zbytočných komplikácií, čo výrazne prispieva ku kvalite a presnosti získaných údajov.

1. **Integrácia hodnotiacich metrík do priebehu testovania:** Pri návrhu hodnotenia sme sa rozhodli postupovať tak, aby testovanie prebiehalo optimálne podľa práce 2.3, a to:
 - (a) **súčasne s plnením úloh,**
 - (b) **bezprostredne po ich ukončení.**
2. **Plynulý prechod na hodnotenie:** Tento aspekt sa zameriava na elimináciu zbytočných operácií po ukončení testovaného scenára a pred dosiahnutím fázy odovzdania výsledkov na ďalšie spracovanie. Príkladom zbytočných krokov je potreba vykonávania dodatočných akcií, ako napríklad v prípade použitia Google Forms, opísaného vyššie. Cieľom je, aby testovaný subjekt bol presmerovaný na hodnotiacu sekciu prostredníctvom jednoduchého príkazu (tzv. "one-button" prístup), ktorý vyžaduje minimálny počet interakcií.
3. **Minimalizácia narušenia zážitku:** Tento aspekt sa zameriava na schopnosť NDMVR zachovať pozitívny zážitok používateľa, či už prostredníctvom prehľadného pozadia alebo čiastočného zachovania pôvodného pozadia scény. Cieľom je, aby sa používateľ stále cítil integrovaný do prostredia a scenára, pričom jeho zážitok nie je narušený prechodom do hodnotiacej sekcie.
4. **Samostatnosť v interakciách:** Cieľom je navrhnúť testovanie a jeho prostriedky tak, aby používateľovi umožnili plne interagovať a vykonávať úlohy nezávisle, bez potreby častých asistencií.

Ďalej sa budeme predovšetkým venovať bodu číslo 1. Body číslo 2 a 3 sú podrobnejšie opísané a zabezpečené z hľadiska návrhu a implementácie v časti 4.2.

5.1.1 Integrácia hodnotiacich metrík do priebehu testovania

Pre dosiahnutie adekvátnych výsledkov na základe analýzy 2.3 je nevyhnutné definovať testovací scenár a správne nastaviť hodnotiace metriky pre jednotlivé úlohy, ako aj pre celkové vyhodnotenie zážitku. Tento prístup je všeobecne opísaný v požiadavkách uvedených v bode 1 v sekcii 5.1, konkrétne:

a. Súčasné získavanie výsledkov počas priebehu testovania:

Tento bod sa zameriava na sledovanie časového intervalu vykonávania úlohy v rámci poskytnutého scenára. Bude hodnotiť náročnosť jednotlivých úloh, ako aj ich vzťah k ďalším úlohám. Zameriame sa aj na intuitívnosť, najmä pri prvej úlohe, a efektivitu učenia sa, nakoľko scenár by mal pozostávať z iterácií krokov potrebných na základnú analýzu fyzikálnych dát získaných z experimentu ALICE. Predpokladáme, že tieto kroky budú často využívať spoločné charakteristiky, ovládacie prvky a princípy, čo umožní sledovanie krivky učenia sa v systéme NDMVR.

b. Získavanie výsledkov po priebehu testovania:

Zameriavame sa na celkové hodnotenie zážitku bezprostredne po ukončení úloh, pričom výsledky budú analyzované a vyhodnotené v rámci celkového zážitku. Pre tento účel sme si zvolili metriku summatívneho merania angažovanosti (UES - User Engagement Scale), opísanú v časti 2.5. Dôležité bolo tiež zvážiť dĺžku tejto metriky na sledovanie výsledného zážitku a jeho faktorov, nakoľko testovanie by nemalo presiahnuť stanovený časový limit maximálne 10 minút na respondenta. Ide o výstavu zameranú na 70. výročie CERN-u na Slovensku. Naše testovanie, ktoré zahŕňa prezentáciu prostredia NDMVR a definovanie základných požiadaviek na dobrovoľných respondentov, je navrhnuté tak, aby pre účastníkov nepredstavovalo nadmernú záťaž. Testovanie by malo byť skôr príjemným úvodom do systému, ktoré poskytne spätnú väzbu bez nadmerného množstva otázok a úloh. Na základe týchto požiadaviek sme zvolili kratšiu formu tejto metriky (UES Short Form), ktorá je uvedená spolu s jej obsahom v časti 2.5.2.

5.1.2 Detailný popis krokov a metodiky testovania

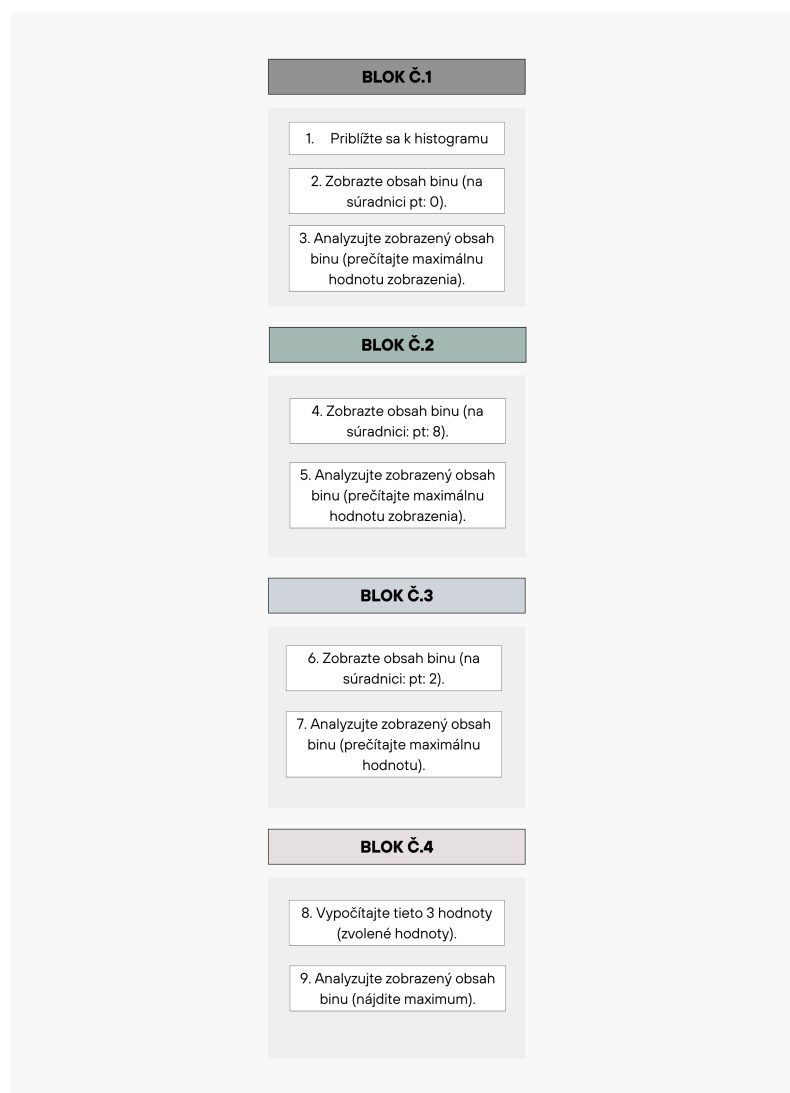
Testovanie bude prebiehať v 3 etapách:

1. **Etapa zameraná na respondenta:** V etape č.1 sa pokúsime získať čo najviac informácií, pričom sa vyhneme zhromažďovaniu citlivých údajov o respondentoch. Pre účely efektívnejšej analýzy v budúcnosti, najmä pri sledovaní výsledkov a ich korelácii k parametrom informatívneho dotazníka respondenta, budeme tieto údaje zbierať prostredníctvom pripraveného PC zariadenia, kde respondent sám vyplní nami vytvorený dotazník ¹. Dotazník bude obsahovať rôzne škály odpovedí, napríklad veková škála, škála dosiahnutého vzdelania, škála vyštudovaného odboru, prípadne skúsenosti s fyzikálnou analýzou alebo s VR zariadením. Tento prístup bude najefektívnejší z hľadiska počtu návštevníkov, ktorí prechádzajú daným stanoviskom.
2. **Etapa plnenia úloh v rámci testovacieho scenára:** Etapa č. 2 je zameraná na súčasné získavanie výsledkov počas priebehu testovania NDMVR. Respondent obdrží len nevyhnutné informácie potrebné na ovládanie a prácu v prostredí VR, ako aj niekoľko minút na získanie istoty v danom headsete a jeho zobrazení. Následne bude oboznámený so scenárom, ktorý bude mať k dispozícii opäť prostredníctvom už zmieneného prístupu v bodoch 2 a 3 časti 5.1. Po jeho zobrazení bude vykonávať jednotlivé úlohy uvedené v scenári testovania. Tieto úlohy sú rozdelené do 4 blokov - väčších úloh. Po vypracovaní úlohy 1 bude čas bežiaci na pozadí zastavený a uložený, to isté platí aj pre nasledujúce úlohy.
3. **Etapa hodnotenia angažovanosti** Etapa č. 3 sa zameriava na bezprostredné hodnotenie angažovanosti a jej faktorov (estetická príťažlivosť, vnímaná použiteľnosť, faktor odmeny, sústredená pozornosť) po ukončení scenára. Výsledky sú zaznamenané formou dotazníka v prostredí NDMVR. Po vyplnení všetkých otázok budú výsledky odoslané na úložisko a po ich obdržaní, bude v prípade záujmu respondenta priestor na krátku diskusiu, počas ktorej môže vyjadriť svoje ďalšie postrehy, ktoré neboli zahrnuté v dotazníku alebo neboli povedané počas plnenia úloh. Všetky relevantné detaily, ako aj reakcie účastníkov počas plnenia úloh alebo po ich ukončení – či už vo forme verbálnych odpovedí na položené otázky, alebo prostredníctvom viditeľnej interakcie počas testovania – budú zaznamenané a následne analyzované za účelom zlepšenia prostredia NDMVR.

¹<https://forms.gle/MV7CQcqEsyH4ss5w6>

5.2 Scenár testovania

Scenár testovania zobrazený na obr. 5.1 je rozdelený do štyroch blokov. Bloky 1, 2 a 3 sú zamerané na analýzu údajov prostredníctvom interakcie s histogramom a jeho binmi, pričom cieľom je testovať schopnosť respondenta intuitívne a efektívne pracovať s dátami. Blok 4 sa zameriava na perspektívne využitie systému NDMVR pri analyzovaní binov histogramu získaných z krokov v predchádzajúcich blokoch. Následne sa tieto údaje kalkuluju pomocou špecifickej funkcie, ktorá poskytne prehľad hmotnostných hodnôt, pričom respondent analyzuje a interpretuje výsledky.



Obr. 5.1: Scenár testovania

Detailnejší opis etáp a ich jednotlivých krokov:

Blok č.1 - Analytická časť: Táto časť je zameraná na skúmanie intuitívnosti ovládania a schopnosti respondenta reagovať na pohybové akcie bez vážnych problémov, ako je kinetóza (závraty alebo nevoľnosť pri interakcii v prostredí VR). Krok zameraný na pohyb sleduje, ako respondent dokáže vykonávať pohybové akcie v prostredí VR a interagovať s ním bez fyzických alebo psychických nepríjemností. Zobrazenie obsahu binu sa v tejto časti sústreďuje na prvý krok procesu, kde respondent vizualizuje údaje a analyzuje ich prostredníctvom interakcie s jednotlivými binmi histogramu. Táto úloha tiež sleduje čitateľnosť zobrazených údajov na plátne a schopnosť orientácie v poskytovanom grafickom zobrazení. Respondent odčíta zadanú hodnotu, pričom sa hodnotí jeho schopnosť interpretovať údaje, čo je kľúčové pre presné vykonávanie úloh vyžadujúcich analýzu výsledkov.

Blok č.2 - Pokročilá analýza: Sleduje zručnosť interakcie a reakcie na zmeny údajov. Zobrazenie binu na stanovenej súradnici (pt: 8) sa sústreďuje na ďalšiu časť histogramu, kde respondent skúma rôzne vizualizácie údajov v rôznych bodoch. Sleduje sa schopnosť respondenta orientovať sa v histograme a jeho zobrazeniach. Nasleduje krok zameraný na reakciu na konkrétnu hodnotu v histograme, konkrétne maximálnu hodnotu, ktorá sa zobrazuje na plátne. Sleduje sa, ako respondent dokáže spracovať zmeny oproti zobrazenému binu z bloku č.1 a identifikovať konkrétne parametre.

Blok č.3 - Pokročilé sledovanie: Cieľom tejto časti je sledovanie rovnakých vlastností ako v bloku č.2, s dôrazom na zistenie časového intervalu učenia sa a orientácie v prostredí.

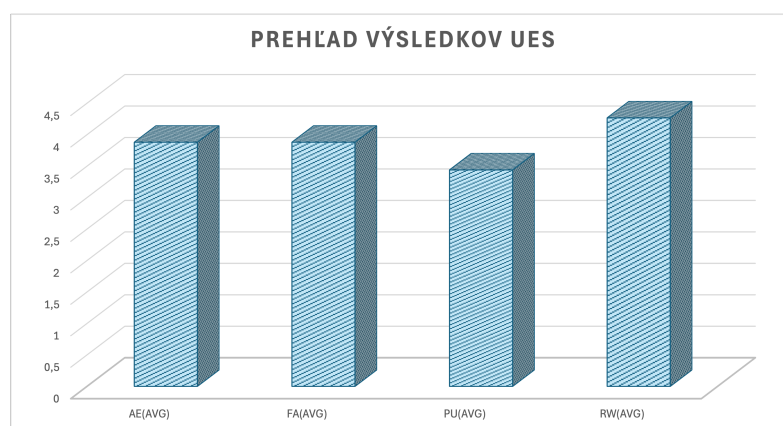
Blok č.4 - Vyhodnotenie: Tento blok je zameraný na presnosť a spoľahlivosť výpočtov, ako aj na prezentáciu potenciálu navrhovaného riešenia. Sleduje schopnosť respondenta nielen vyvodiť závery na základe zobrazených údajov, ale aj intuitívne používať analytické nástroje, ktoré tieto údaje spracúvajú. Hodnotí sa efektivita identifikácie maximálnych hodnôt uvedených hmotností častíc a ich interpretácia v kontexte celkových výsledkov.

5.3 Záverečné hodnotenie výsledkov

V tomto hodnotení výsledkov zohľadňujeme angažovanosť meranú na základe metodiky UES-SF, ktorá je podrobne opísaná v časti 2.5.2. Výsledky merania sú graficky znázornené prostredníctvom grafov, ktoré zobrazujú výsledky jednotlivých respondentov a priemernú hodnotu testovania. Hodnotenia sa zúčastnilo 25 respondentov rôznych vekových kategórií – od žiakov základných škôl až po

staršie ročníky. Táto časť sa primárne zameriava na vyhodnotenie angažovanosti po testovaní, ako aj na schopnosť učenia sa a reakcie testovaných subjektov na opakujúce sa úlohy v systéme NDMVR. V rámci tohto hodnotenia porovnávame aj už spomínané meranie uvedené v práci [7], ktoré označujeme ako finálne meranie, resp. meranie verzie B. Naopak, meraním verzie A označujeme testovanie realizované v rámci tejto práce.

5.3.1 Hodnotenie celkovej angažovanosti



Obr. 5.2: Celkový prehľad výsledkov UES

Celkový priebeh merania angažovanosti je znázornený na obrázku 5.2, ktorý zachytáva vývoj celkovej angažovanosti. Táto pozostáva zo štyroch hlavných oblastí: vnímanej použiteľnosti (PU), estetickej príťažlivosti (AE), faktora odmeny (RW) a sústredenej pozornosti (FA).

Testovaná verzia dosiahla skóre celkovej angažovanosti **3,87**, pričom výsledky sa pohybovali v rozsahu od 1 do 5, kde 1 predstavuje minimálnu a 5 maximálnu angažovanosť. Na základe tohto skóre môžeme testovanú verziu kategorizovať ako „dobrú“. V predchádzajúcom testovaní prostredia NDMVR, opísanom v práci [7], dosiahla verzia hodnotu 4,29, čo bolo klasifikované ako „veľmi dobrá“ úroveň angažovanosti.

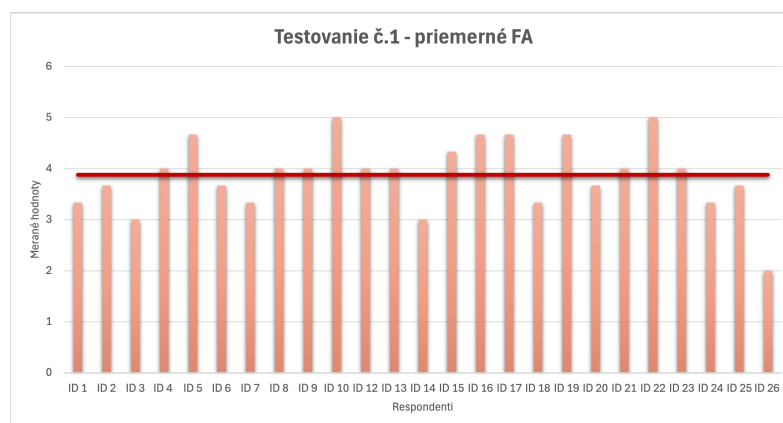
Tento mierny pokles môže byť spôsobený testovaním bez priameho porovnania dvoch verzií, ako aj širšou medializáciou a dostupnosťou VR technológie, čo mohlo ovplyvniť vnímanie angažovanosti. Predovšetkým však zohrali úlohu faktory ako vyšší počet respondentov, lepšia príprava prostredia a scenára, ako aj celkové zachovanie zážitku. Preto tento pokles považujeme za očakávaný a žiaduci, keďže zvyšuje presnosť merania a poskytuje spoľahlivejšie hodnoty.

Hodnotenia sa zúčastnil väčší počet respondentov, čo prispieva k zvýšeniu spomínanej presnosti výsledkov. Je však potrebné zohľadniť aj možné skreslenie

tejto metódy hodnotenia, spôsobené zapojením rôznorodých skupín respondentov, nielen expertov alebo fyzikov. Výsledky môžu byť tiež ovplyvnené širokým vekovým rozptylom respondentov, najmä na spodnej a hornej hranici vekového spektra. Napriek tomu testovaná verzia vykazuje solídnu úroveň angažovanosti.

5.3.2 Hodnotenie vnímanej použiteľnosti

Prvou skúmanou oblasťou bola vnímaná použiteľnosť, ktorá zahŕňa interakciu, mieru kontroly a vynaložené úsilie pri používaní systému. Výsledky sú znázornené na obrázku 5.3. Testovaná verzia dosiahla skóre **3,88**, čo zaraďujeme do kategórie „dobrá použiteľnosť“. V predchádzajúcej práci však finálna verzia B dosiahla skóre **4,49**, teda „veľmi dobrá“. Napriek tomuto poklesu predpokladáme, že aktuálne získaná hodnota je presnejšia vďaka rozsiahlejšiemu a optimalizovanejšiemu testovaniu.



Obr. 5.3: Priemerná vnímaná použiteľnosť (FA-UES) získaná testovaním

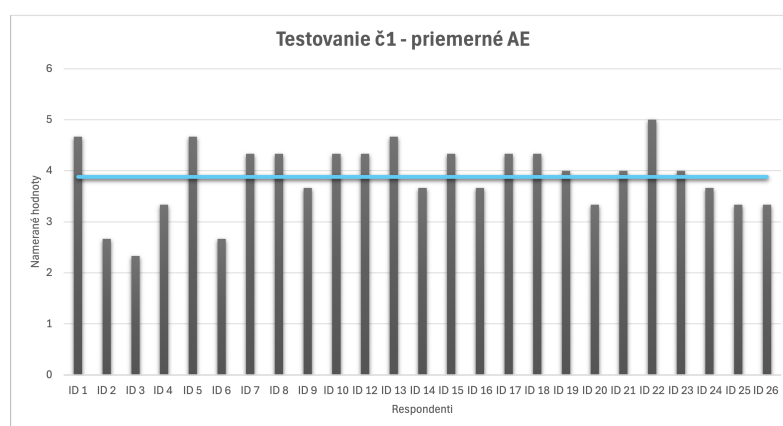
5.3.3 Hodnotenie estetickej pritažlivosti

Druhou skúmanou oblasťou bola estetická pritažlivosť, ktorá zahŕňa vizuálnu kvalitu interiéru a rozhrania. Výsledky zobrazené na obr. 5.4 ukazujú hodnotu 3,88, čo opäť zodpovedá kategórii „dobrá“. Oproti predchádzajúcemu meraniu (4,63 – „veľmi dobrá“) zaznamenávame pokles, ktorý však môže byť výsledkom realistickejšieho testovania a menej skreslených údajov.

Dôležité je tiež zohľadniť nefunkčnosť tabletu, ktorý je neustále k dispozícii používateľom a výrazne ovplyvňuje poskytovanú spätnú väzbu. Jeho funkčnosť zároveň otvára ďalšie estetické možnosti a príležitosti pre používateľa. Jeho estetická stránka bola takmer nulová – obsahoval len základné farebné prvky VR

priestoru, prevažne jednoduché obdĺžniky bez vizuálnej komplexnosti. Po zohľadnení nedostatkov tabletu a vykonanej redukcii rušivých prvkov scény, pričom boli zachované len podstatné elementy VR prostredia počas testovania, považujeme dosiahnutý výsledok za veľmi solídny.

Domnievame sa však, že po odstránení predovšetkým nedostatkov tabletu a zlepšení jeho funkcionality by mohol byť dosiahnutý výsledok ešte vyšší. Na základe reakcií respondentov usudzujeme, že výsledné skóre bolo v značnej miere ovplyvnené, respektíve „zachránené“, prvkom UI menu, ktorý sme implementovali. Tento prvok zároveň slúžil ako stabilný oporný bod počas testovania a používania. Preto tento výsledok považujeme za uspokojivý.



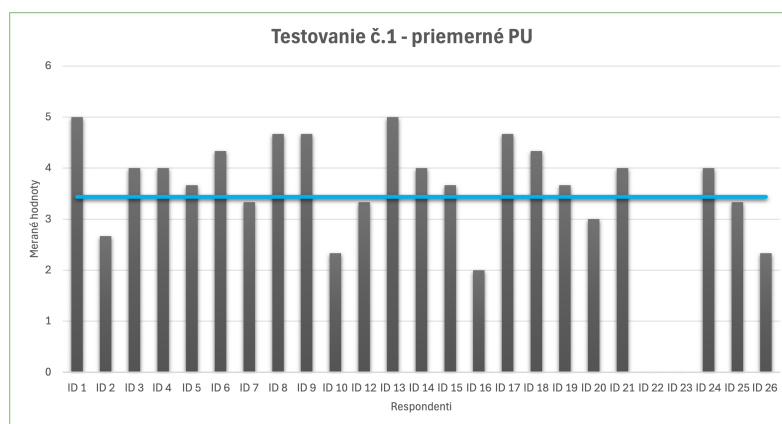
Obr. 5.4: Priemerná estetická pritažlivosť (AE-UES) získaná testovaním

5.3.4 Hodnotenie sústredenej pozornosti

Treťou skúmanou oblasťou bola sústredená pozornosť, ktorá zahŕňa pocity pohltenia do interakcie a stratu vnímania času pri používaní aplikácie. Výsledky sú zobrazené na obrázku 5.5.

Prekvapením bolo, že respondenti vykazovali rovnakú úroveň 3,55 sústredenej pozornosti ako v práci [7] pri verzii B (finálnej verzii).

V priebehu testovania bola tentoraz v priemere zaznamenaná nižšia komunikácia s osobou vykonávajúcou dohľad ako pri verzii B, keďže zásah zo strany dohliadajúcej osoby nebol potrebný. Naopak, snahou bolo minimalizovať akékoľvek zásahy s cieľom objektívne sledovať zmenu výsledkov. Domnievame sa, že k tomuto zlepšeniu prispelo aj UI menu, ktoré respondentom poskytovalo prehľad o postupe úloh a očakávanom priebehu testovania v podobe scenára. Možnosť postupného plnenia úloh a zaznamenávania zmien pravdepodobne prispela k zvýšeniu sústredenia a neprerušeniu zážitku pri post-testovom získavaní spätnej väzby.



Obr. 5.5: Priemerná sústredená pozornosť (PU-UES) získaná testovaním

Ďalším významným faktorom bolo zjednodušenie scény a jej príprava na testovanie – všetky nepotrebné prvky boli odstránené, pričom zostali iba tie, ktoré boli nevyhnutné pre účely výskumu. Tento prístup sa pozitívne odrazil na samotnom priebehu testovania a reakciách respondentov, ktorí sa v systéme lepšie orientovali, rýchlejšie pochopili jeho fungovanie a skúmané aspekty. Na rozdiel od verzie B, kde podľa predchádzajúcej štúdie [7] respondenti často pôsobili mierne zmätene a nepochopili presný význam ani jednoduchého scenára, sa pri testovanej verzii A prejavila lepšia orientácia v prostredí.

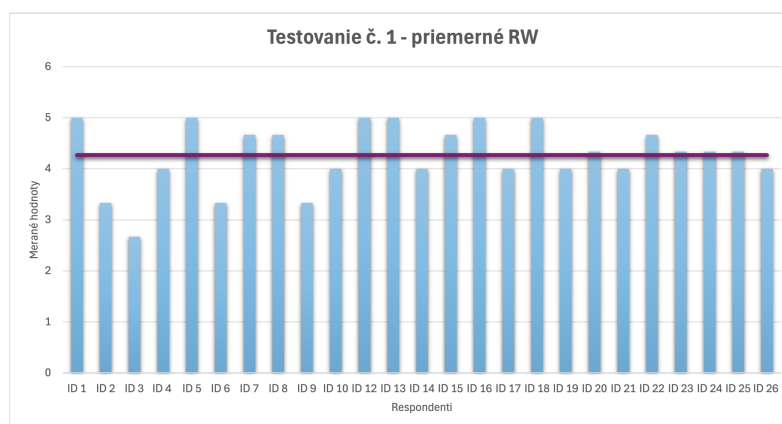
Okrem toho verzia A nebola testovaná spolu s verziou B, čo znamená, že respondenti strávili v systéme a celkovo v technológii VR menej času. Tento faktor mohol mierne ovplyvniť ich sústredenú pozornosť.

5.3.5 Hodnotenie faktoru odmeny

Štvrtou skúmanou oblasťou bol faktor odmeny, ktorý zahŕňa pocity príjemných a uspokojivých zážitkov z interakcie s aplikáciou, ako aj skúmanie ochoty používateľov odporučiť aplikáciu ostatným a angažovať sa s ňou aj v budúcnosti. Merania a ich výsledky sú zobrazené na obrázku 5.6.

Celkový faktor odmeny pre verziu A bol **4,26**, čo je kategorizované ako veľmi dobré hodnotenie. Verzia B priniesla faktor odmeny s hodnotou **4,49**, čo tiež môžeme definovať ako veľmi dobré hodnotenie. Respondenti považovali verziu A za príjemnú a uspokojivú z pohľadu interakcie s aplikáciou, aj napriek problémovým funkcionalitám tabletu, ktoré sú opísané v kapitole 6 venovanej odstráneniu nedostatkov testovania.

Výsledok nás príjemne prekvapil, nakoľko hodnotu považujeme za bližšiu skutočnosti, a verzia A mohla taktiež zaujať respondentov natoľko, že by ju odporučili svojim známym a kolegom. Taktiež to poskytuje dobrú spätnú väzbu o



Obr. 5.6: Priemerný faktor odmeny (RW-UES) získaný testovaním

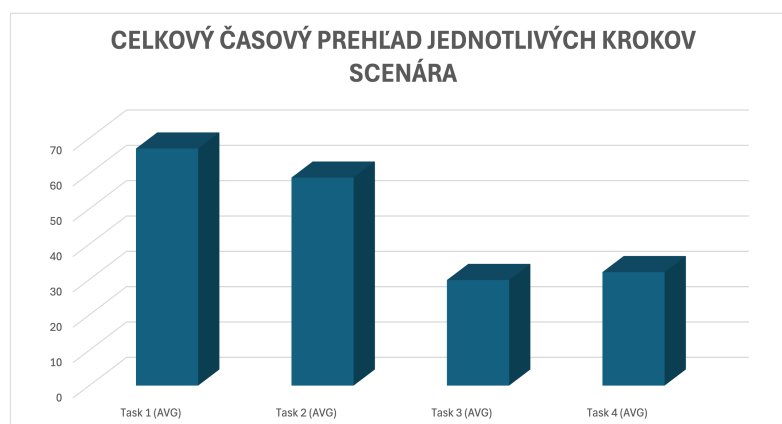
funkcionalite UI menu, nakoľko interakcia s ním bola najintenzívnejšou časťou používania systému. Implementácia, ako je opísané v časti 4.2, môže byť považovaná za veľmi úspešnú, aj napriek menším problémom a nedostatkom tohto prvotného testovania.

Presnejšie skóre verzie A, ktoré sa približuje k verzii B, naznačuje, že respondenti by sa radi s aplikáciou angažovali opäť, čo je veľmi dôležitý faktor z pohľadu udržateľnosti používateľov systému. To platí aj napriek pomerne dôležitej a zásadnej chybe, ako sú reakcie a estetická stránka tabletu. Tento výsledok nás utvrdzuje v tom, že po jeho odstránení a nahradení lepšou verziou by sa toto skóre mohlo v budúcnosti zlepšiť.

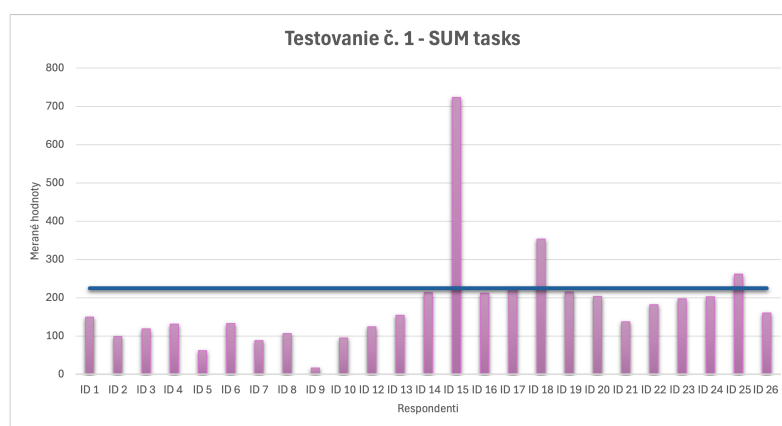
5.3.6 Vyhodnotenie jednotlivých krokov scenára

Dané vyhodnotenie sleduje časový vývoj jednotlivých blokov (4 sekcie, do ktorých bol rozdelený scenár testovania) uvedený v časti 5.1. Každý blok, pozostávajúci z menších podúloh, bol zobrazený v UX menu v podobe šablóny „to-do list“, kde po dokončení bloku používateľ označil jeho splnenie, čím sa ukončil čas merania daného bloku. Priemerný čas vykonávania scenára je znázornený na obrázku 5.8.

Výsledky prezentované na obr. 5.7 zobrazujú priemerný čas jednotlivých blokov. Ukazujú, že najdlhšie trvalo vykonávanie úvodného bloku, ktorý predstavoval prvú inicializačnú fázu z pohľadu interakcie a práce so systémom. Forma bloku bola rovnaká ako v blokoch 2 a 3, čo možno vysvetliť znižujúcou sa časovou závislosťou a lepšou vykonateľnosťou. Krivka učenia sa zlepšovala v podobe klesajúceho času. Trvanie bloku 3 kleslo o takmer polovicu v porovnaní s blokom 2, zatiaľ čo blok 2 klesol len o približne jednu štvrtinu v porovnaní s blokom 1 (pozri obr. 5.7).



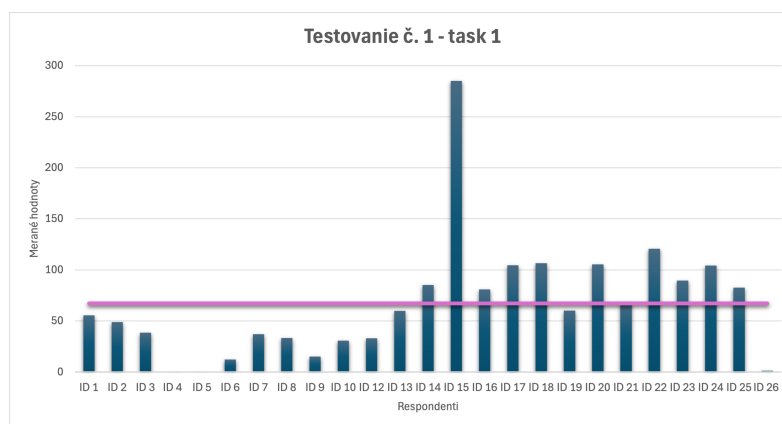
Obr. 5.7: Celkový prehľad časových intervalov jednotlivých blokov scenára testovania



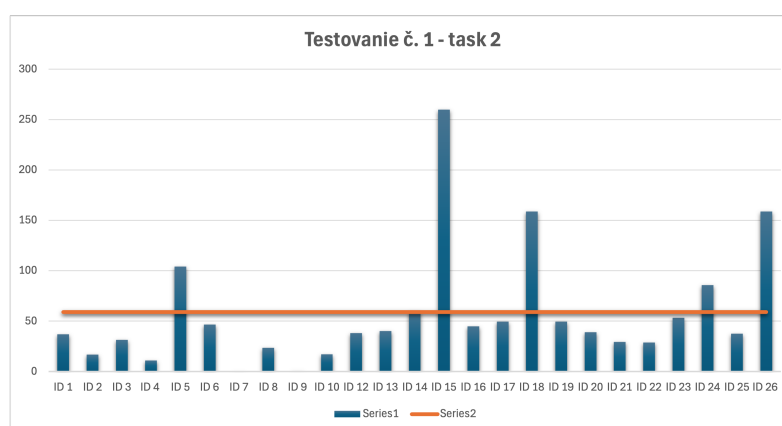
Obr. 5.8: Priemerný čas vykonávania scénara testovania

Pokles času potrebného na vykonanie blokov počas testovania naznačuje, že používatelia sa čoraz viac oboznamujú so systémom, čo vedie k efektívnejšiemu a rýchlejšiemu plneniu úloh. Z pohľadu zvládnutia systému to znamená, že po úvodnom zoznámení sa s jeho funkciami sa interakcia stáva plynulejšou, čo skracuje čas potrebný na dokončenie jednotlivých blokov. Tento trend ukazuje, že aj keď môže byť systém na začiatku náročnejší, s pribúdajúcimi skúsenosťami sa stáva intuitívnejším a používateľsky prívetivejším, čo je kľúčové pre dlhodobú efektívnosť a udržateľnosť systému. Podobné zistenia sú podporené aj výskumom [20] v oblasti učenia sa a získavania znalostí.

Blok č.4 obsahoval časť vyhodnotenia a analýzy zvolených binov, kde bola časová odozva mierne vyššia v dôsledku zmeny formy úlohy a času potrebného na nájdenie a použitie možností systému pre túto analýzu. Tento čas bol však stále podstatne kratší ako počiatočný čas prvej úlohy, čo hodnotíme ako pozitívny výsledok. Priemerný čas trvania uvedený na obr. 5.9 pre **blok č.1** bol **67 sekúnd**, čo je najvyššia hodnota v rámci daného scenára testovania.

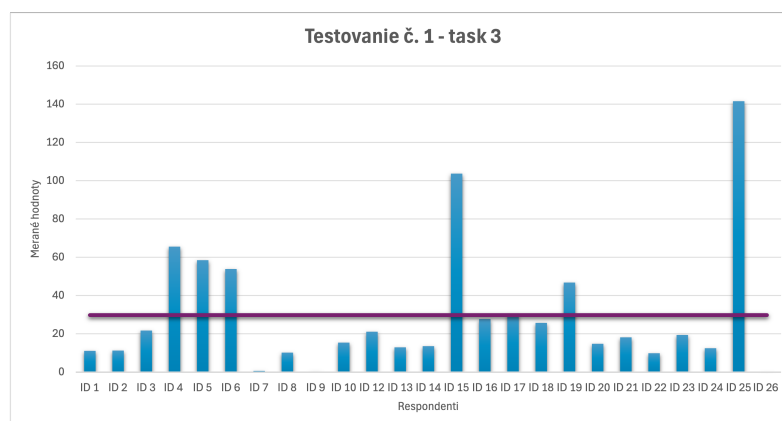


Obr. 5.9: Priemerný čas pri riešení bloku č.1 scenára testovania



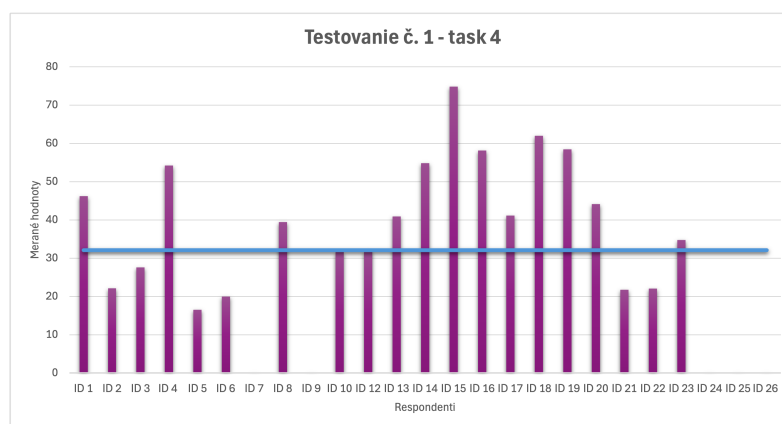
Obr. 5.10: Priemerný čas pri riešení bloku č.2 scenára testovania

Blok č. 2, uvedený na obr. 5.10, mal nameraný časový interval **58 sekúnd**. **Blok č. 3**, znázornený na obr. 5.11, trval **29 sekúnd**. Forma vyhodnotenia týchto dát, uvedená na obr. 5.12, pre **blok č. 4** vyžadovala **32 sekúnd**.



Obr. 5.11: Priemerný čas pri riešení bloku č.3 scenára testovania

Priemerný celkový čas vykonania scenára uvedený na obr. 5.8 bol **186 sekúnd**, čo po zaokrúhlení smerom nadol predstavuje **3 minúty**.



Obr. 5.12: Priemerný čas pri riešení bloku č.4 scenára testovania

Výsledky nemusia byť úplne presné, keďže istý počet respondentov nedokončil daný scenár až do konca, alebo ich záznamy neboli dokončené a odoslané. Tak tiež niektorí respondenti poskytli spätnú väzbu iba ústnou formou, bez úplného dokončenia úloh. Predovšetkým išlo o hodnoty vekovej kategórie, ktoré smerovali buď nadol, alebo nahor.

5.3.7 Závěry hodnotenia

Výsledky uvedené vyššie podporili sumatívne hodnotenie, teda kvantitatívne hodnotenie UX. Konkrétne zmeny v závere ovplyvnila aj forma interakcie.

Z 25 respondentov až 20 (teda nadpolovičná väčšina) prejavilo frustráciu z umiestnenia tabletu, jeho veľkosti, ako aj z nedostatočného dizajnu a nespoľahlivej funkcionality. Zaznamenané boli aj nedostatočné renderovacie schopnosti technológie A-Frame – podľa odpovedí respondentov obraz nebol taký kvalitný, ako očakávali, alebo ako už videli či skúšali v iných prípadoch. Tento nedostatok mohla umocniť aj nekonzistentnosť vizuálneho zobrazenia – namiesto jednej kvalitnej schémy bolo podporovaných viacero variantov, čo mohlo narúšať profesionálny dojem.

Toto tvrdenie však vyvrátilo 15 respondentov, ktorí síce vyjadrili frustráciu z tabletu, no oceňovali možnosti, ktoré ponúkal. Najmä mladšie ročníky ocenili možnosť prispôsobenia scén a ďalších vizuálnych aspektov histogramu.

Negatívne na väčšinu respondentov pôsobili aj biny (stĺpce histogramu), ktoré sú v NDMVR vizuálne reprezentované primitívnymi kockami (boxmi). Na otázku, či by ich úprava tento problém vyriešila, odpovedali nejednoznačne.

Naopak, možnosti ovládania a pohybu boli hodnotené prevažne pozitívne a respondenti im v subjektívnom hodnotení nevenovali väčšiu pozornosť. Najviac ocenili zjednodušenú verziu obsahu scény, ktorá im umožnila lepšie pochopiť jej

účel a zároveň komfortnejšie vnímať informácie. V závere bola poskytnutá možnosť ukážok iných plných verzií prostredí a dát. Pozitívne hodnotili aj tzv. „letový režim“, ktorý prispieval k plynulosti a vyváženosti pohybu v prostredí.

UI Menu bolo navrhnuté tak, aby ostávalo v rovnakej vzdialenosti od používateľa aj pri pohybe, keďže bolo umiestnené na kamere pohľadu. Táto vlastnosť, spolu s možnosťou využiť Menu ako osobného asistenta na poskytovanie prehľadu o úlohách a ich odškrtnutie, bola vnímaná veľmi pozitívne. Jeho funkcionality ocenili aj potenciálni používatelia zo zboru Katedry jadrovej fyziky UPJŠ v Košiciach. Testovania sa zúčastnili nielen študenti, ale aj pedagógovia, pričom niektorí boli súčasťou aj predchádzajúceho testovania opísaného v práci [7]. Prítomnosť tejto priamo dostupnej verzie testovacieho scenára bola v porovnaní s predchádzajúcou papierovou alebo čítanou alternatívou hodnotená veľmi pozitívne.

Respondenti taktiež ocenili možnosť prehrávania hudby počas analýzy, čo prispelo k lepšiemu používateľskému zážitku. Táto vlastnosť bola obzvlášť oceňovaná mladšími vekovými kategóriami. Celkovo hodnotili ovládanie prostredia ako intuitívne, s výnimkou tabletu, ktorý si podľa nich vyžadoval určité úpravy. Nadpolovičná väčšina respondentov uviedla, že by bolo vhodné prepracovať jeho dizajn a umiestnenie. Niekoľko z nich dokonca vyjadrilo názor, že tablet by očakávali ako súčasť UI Menu.

Podľa väčšiny bolo používanie v rámci testovacieho scenára opísaného v časti 5.1 jednoduché a intuitívne z hľadiska analýzy údajov.

Závery vyplývajúce z poskytnutých vyjadrení:

- Reorientácia funkcionality tabletu a jeho dizajnu.
- Odhalené nedostatky v UI Menu:
 - Opätovné odosielanie tých istých údajov pri rýchlom kliknutí
 - Možnosť získavania odpovedí vo formáte CSV pre rýchlejšie spracovanie a vyhodnotenie výsledkov.
 - Nedostatočná viditeľnosť textu a jeho prekryvanie pri určitých podmienkach, najmä pri pohybe a otáčaní hlavou.
- Zváženie migrácie na inú technológiu na zobrazovanie obsahu.
- Prehodnotenie spôsobu vizualizácie binov.
- Doplnenie novej scény s profesionálnejším vzhľadom.

6 Nedostatky odhalené v rámci testovania a ich odstránenie

Táto kapitola sa zameriava na odstránenie väčšiny nedostatkov odhalených počas testovania, opísaného v kapitole 5. Na základe týchto zistení navrhujeme konkrétne zlepšenia, ktoré odstránia identifikované problémy, a detailne popíšeme implementáciu nového riešenia vrátane jeho nasadenia.

6.1 Migrácia funkcionalít VR tabletu

Táto časť sa venuje analýze súčasného riešenia NDMVR tabletu. Hlavným cieľom je detailne preskúmať jeho architektúru, identifikovať kľúčové nedostatky a ich príčiny, porovnať ju s predchádzajúcou verziou a využiť pritom relevantné výsledky experimentu uvedeného v časti 5.3.7.

6.1.1 Analýza použitých riešení tabletu NDMVR

V aktuálnej verzii NDMVR je k dispozícii aktualizovaná verzia tabletu vychádzajúca z práce bývalého študenta. Táto verzia priniesla jednotný JSON konfiguračný súbor, ktorý je poskytovaný službou **ndmvr redirect** na účely zdieľania konfigurácie medzi jednotlivými komponentmi a jej následné spracovanie. Významnou zmenou oproti práci [7], ktorá sa taktiež zaoberala problematikou VR tabletu, bola zmena starej verzie a pokračovanie v návrhu na novej architektúre. Táto architektúra zahŕňala samostatný komponent pre desktopové zobrazenie a samostatný komponent pre zobrazenie vo VR.

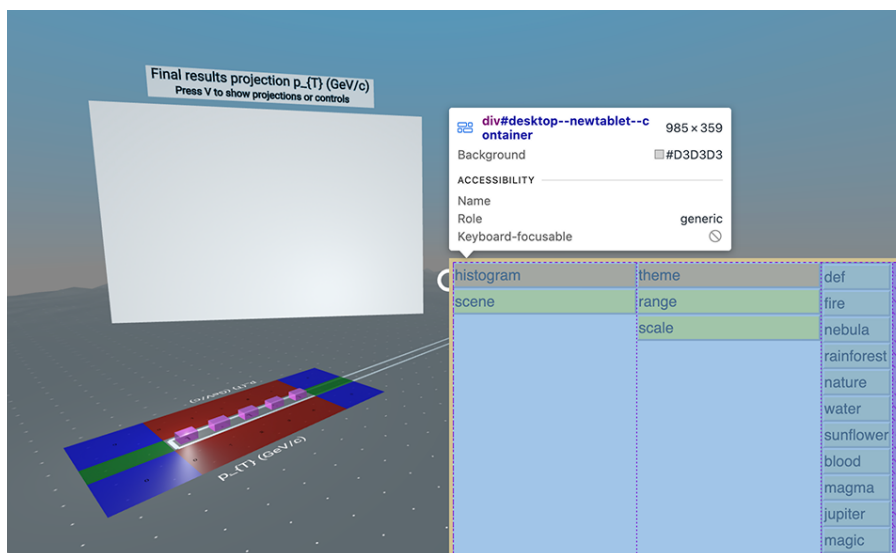
Implementácia bola založená na vtedajšej funkcionalite systému **ndmvr context**, v ktorom objekt state slúžil na uchovávanie stavu a jeho následné spracovanie. Aktuálna verzia tabletu priniesla niekoľko zlepšení, ako zavedenie jednotného komponentu, nahradenie state objektu mechanizmom redirect, a tiež render podmienený rozšíriteľnou konfiguráciou. Napriek týmto vylepšeniam obsahovala stále niekoľko závažných nedostatkov.

6.1.2 Odhalené nedostatky aktuálneho riešenia a ich dôvody

Medzi hlavné problémy patrila nefunkčnosť tabletu po interakcii so scénou v prostredí A-Frame. Tento problém bol spôsobený nesprávnym použitím služby ndmvr redirect a nevhodnou implementáciou renderovania tabletu – v desktopovom režime pomocou HTML prvkov a v headsetovom zobrazení len prostredníctvom základných entít A-Frame.

Ďalším nedostatkom bolo nesprávne využitie primitív VR scény, najmä chybné implementovaný raycaster určený na detekciu používateľských akcií v trojrozmernom priestore. Kvôli tomu neboli niektoré akcie správne zachytené a veľkosť interakčnej plochy nezodpovedala reálnym podmienkam vo VR ako je opísané v časti 6.1.3.

Hlavným dôvodom nefunkčnosti tabletu bola strata konfigurácie v redirecte, ktorá sa neaktualizovala podľa očakávaného scenára a zvolenej akcie. Tento problém vznikol v dôsledku prepísania obsahu, čím sa narušil plynulý tok operácií. Keďže v desktopovom režime bol tablet renderovaný nad scénou prostredníctvom HTML (viď obr. 6.1), tento aspekt viedol k strate možnosti interakcie s tabletom po vstupe do VR režimu scény, nakoľko bol renderovaný mimo A-Frame, a teda mimo hlavného zobrazenia.



Obr. 6.1: Viacúrovňové in-line zobrazenie tabletu NDMVR (zdroj: NDMVR)

Okrem tohto zásadného nedostatku implementácia obsahovala aj ďalšie problémy, ako nedostatočnú integráciu interakčných možností, ovládacích prvkov, prispôbenia zobrazenia a vizuálneho dizajnu, čo výrazne ovplyvnilo celkovú použiteľnosť.

Prehnane minimalistické prevedenie dizajnu, či už v headsetovom alebo desk-

topovom režime, ešte viac umocnilo nedostatočnú používateľskú skúsenosť (UX).

6.1.3 Hodnotenie aktuálneho riešenia respondentmi

Na základe odpovedí respondentov získaných v časti 5.3.7 bolo identifikovaných viacero problémov. Okrem nedostatku opísaného v časti 6.1.2, ktorý sa prejavoval najmä častými reakciami na nefunkčnosť tabletu, boli často spomínané aj ďalšie výhrady.

Respondenti opakovane vyjadrovali frustráciu nielen z neintuitívneho umiestnenia, veľkosti a dizajnu tabletu, ale aj z jeho celkového vnímania ako prekážky pri práci so scénou. Mnohí poukazovali na jeho rušivé umiestnenie – súčasná implementácia bola objemná a zasahovala do zorného poľa používateľa, najmä pri vystretej ruke, keď bol tablet neustále viditeľný. Jeden z respondentov ho charakterizoval slovami: „Pripomína to dizajn z 90. rokov.“

Experiment, opísaný v časti 5.3.7, poskytol spätnú väzbu poukazujúcu aj na problém s nepresnou detekciou akcie „klik“ pomocou triggeru na ovládači Meta Oculus 2/Oculus 3. Ďalším identifikovaným problémom bolo, že interakcia prebiehala pravou rukou s tabletom umiestneným na ľavej ruke, čo spôsobovalo neprirodený a nepraktický uhol pre manipuláciu. Tablet bol zároveň relatívne veľký a umiestnený vysoko na ruke, čo výrazne obmedzovalo jeho ergonomickosť a spôsobovalo používateľom nepohodlie. Viacerí respondenti sa nad týmto riešením len pousmiali, čo nepriamo poukazuje na jeho nízku praktickosť. Uvedené faktory prispievali k celkovej nepoužitelnosti tabletu v testovanom prostredí.

6.1.4 Návrh odstránenia nedostatkov prostredníctvom nového riešenia

S cieľom vyriešiť tieto problémy sme sa rozhodli prepracovať aktuálnu verziu tabletu a nahradiť ju riešením navrhnutým v tejto práci, ktoré je podrobne popísané v časti 4.2.

Nová verzia bola implementovaná ako šablóna (template) pre rôzne typy výberových možností, a to výlučne na strane A-Framu pre akékoľvek zobrazenie. Týmto prístupom sa podarilo odstrániť podstatnú časť problémov s interakciou. Pri návrhu sme vychádzali z pôvodnej konfigurácie a jej organizačnej štruktúry, keďže tá bola logická a prehľadná nielen z nášho pohľadu, ale aj z pohľadu autora a hlavného prispievateľa riešenia.

Ponechaná konfigurácia obsahovala hlavnú kategorizáciu (schému) výberu možností, ako napríklad Scene alebo Histogram. V predchádzajúcej verzii bolo

toto rozdelenie implementované ako viacúrovňové „in-line“ zobrazenie (viď obr. 6.1), ktoré podľa spätnej väzby používateľov vyvolávalo frustráciu a zmätok. Tento stav negatívne ovplyvňoval ochotu používateľov experimentovať alebo cielene využívať dostupné možnosti, keďže nebolo zrejmé, čo jednotlivé cesty predstavujú a kde sa v rámci navigácie práve nachádzajú.

Rozhodli sme sa túto kategorizáciu ponechať, avšak presunúť ju na úroveň zmeny celej zobrazovacej plochy okna. Tento prístup umožňuje logické rozdelenie komponentov a ich subkomponentov do skupín, s ktorými používateľ nemusí priamo interagovať. Pohybom do strany sa používateľ dostane na príslušnú úroveň nastavení.

Zobrazenie okna tabletu zabezpečuje implementácia uvedená v časti 4.2, konkrétne prostredníctvom hlavnej zobrazovacej plochy a navigačného panela. V ňom bude daný komponent umiestnený pod sekciou nastavení a reprezentovaný ikonou, ktorá je používateľom známa. Po otvorení tabletu sa zobrazí minimalistické výberové okno s možnosťami na úpravu konkrétnych vlastností prostredia. Po výbere konkrétnej vlastnosti budú následne zobrazené ďalšie možnosti v závislosti od typu výberovej akcie definovanej v konfigurácii (napr. select alebo range).

6.1.5 Implementácia konfigurácie tabletu v rámci UI menu zobrazenia

V rámci implementácie sme sa rozhodli zamerať na zlepšenie adaptability konfigurácie používateľského rozhrania (UI Menu), a teda aj samotného tabletu. Cieľom bolo dosiahnuť vyššiu flexibilitu nielen na úrovni zobrazenia, ale aj z hľadiska možnej migrácie do iných frameworkov, ako napríklad React, opísaný v časti 1.5. Tieto zmeny sú kľúčové pre ďalší rozvoj systému a jeho dlhodobú udržateľnosť.

Nová konfigurácia bola navrhnutá v rámci uvedených zmien v súlade so štandardom Open API ¹ spolu s ostatnými šablónami uvedenými v časti 4.2.5.

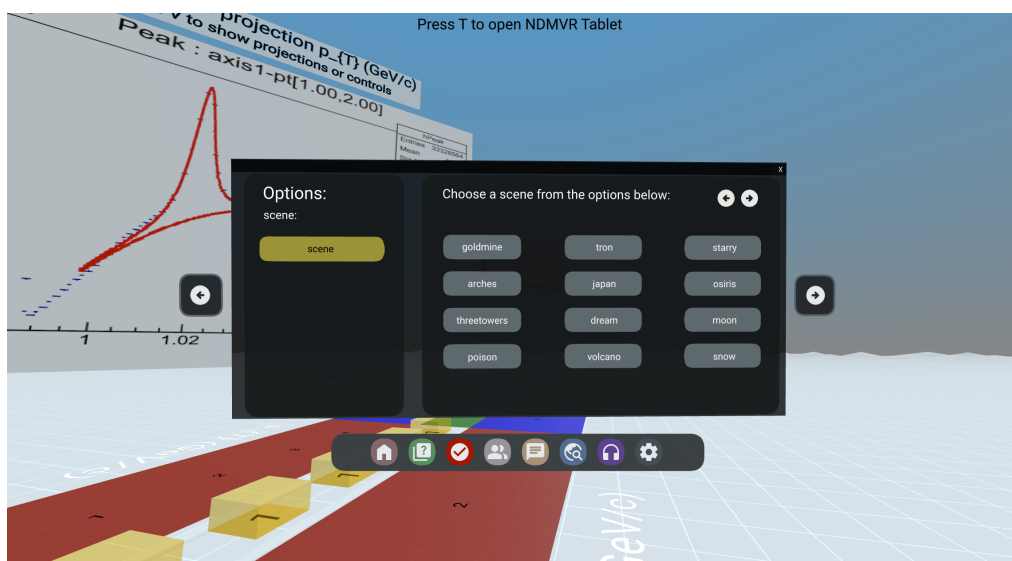
6.1.6 Implementácia konfigurácie a zobrazenia tabletu

Implementácia uvedená v časti 4.2.4 bola obohatená o nový template, pričom selekcia bola zabezpečená prostredníctvom dvoch renderovacích jednotiek. Ľavá strana sa podieľala na výbere a renderovaní jednotlivých vlastností (properties) integrovaných do primárnej konfigurácie, ktorá bola zameraná na nastavenia. Táto časť obsahovala konfigurovateľné atribúty ako properties, description, schema a nastavenie hodnoty (value).

¹<https://swagger.io/specification/>

Každá property v hlavnej schéme nastavení mala svoje špecifické vlastnosti, ako aj ďalšie atribúty, napríklad options a title. Schéma jednotlivých položiek bola sledovaná a diferencovaná pre každú property. Jej renderovanie zabezpečilo samotné okno, ktorého zmeny (napríklad posun do strán) obnovili obsah pre danú schému a načítali všetky príslušné properties definované touto schémou.

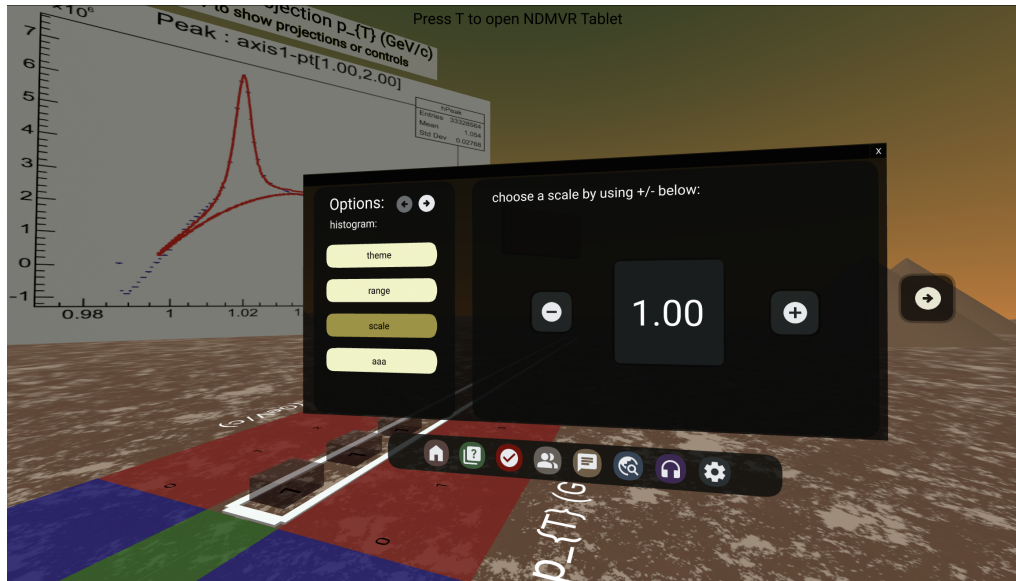
Pri výbere položky sa automaticky aktualizuje zobrazenie na pravej strane, kde sa zobrazí detail vybranej položky. Tento mechanizmus umožňuje intuitívne prechádzanie medzi rôznymi sekciami, pričom menu stále zachováva svoju dynamiku. Pravá strana poskytuje zobrazenie jednotlivých možností pre každú vlastnosť. Funkcionalita je znázornená na obr. 6.2.



Obr. 6.2: Vytvorený template pre typ výber (select), konkrétne výber scény histogramu (zdroj: NDMVR)

Šablóna obsahuje dva typy výberu, ktoré umožňujú manipuláciu s údajmi daného výberu (options). Voľba konkrétneho typu sa uskutočňuje prostredníctvom definície vlastnosti **type** v konfigurácii. Používateľ si môže zvoliť výber označovaný ako select (viď obr. 6.2), ktorý zobrazuje jednotlivé prvky poľa options. Aktiváciou prvku sa vykoná výber, atribút value v konfigurácii sa zmení na príslušnú option a ďalej sa spracuje v rámci komponentu.

Samotné telo komponentu obsahuje preddefinovanú funkcionality, ktorá zabezpečuje aktualizáciu globálnej schémy, napríklad histogramu alebo scény. Pri výbere typu **range** (viď obr. 6.3) je k dispozícii konfigurovateľný interval s definovateľnými krokmi voľby, pričom používateľ môže pomocou tlačidiel upravovať výber smerom k hornej alebo dolnej hranici intervalu.



Obr. 6.3: Ukážka templatu pre typ rozpätia (range), konkrétne výšky binu (zdroj: NDMVR)

6.1.7 Zvýšenie adaptability konfigurácie tabletu a celého UI Menu

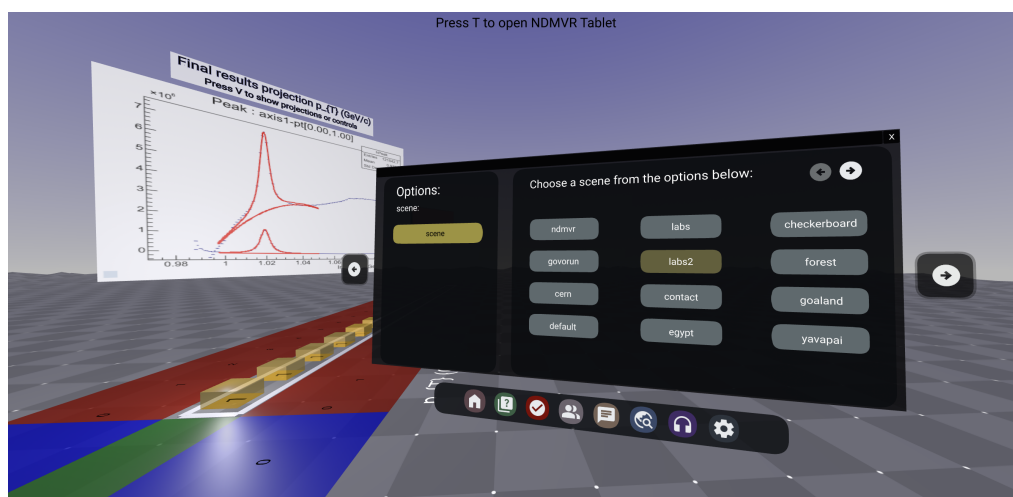
V rámci konfigurácie bola pridaná možnosť definovať počet zobrazených položiek a správanie pri prechode cez paginator. Tento krok výrazne zlepšuje používateľský zážitok pri interakcii s menu a poskytuje flexibilitu pri spracovaní rôznych šablón.

Každá vlastnosť (property) v rámci šablóny môže byť priradená k špecifickej akcii. Ak pre danú vlastnosť nie je akcia definovaná v konfigurácii, automaticky sa vykoná predvolená operácia. Pri konfigurácii nastavení tabletu sa emitujú udalosti (events), ktoré umožňujú okamžitú aktualizáciu globálnej schémy, ako je histogram alebo scéna. Napríklad v prípade dotazníkového formulára sa odpovede odosielajú na preddefinované úložisko, zatiaľ čo pri monitorovaní postupnosti aktivít sa zaznamenáva čas vykonania a údaje sa po splnení úlohy ukladajú.

Pri implementácii prehrávača sa skladby načítavajú zo zadaných URL adries, ktoré odkazujú na zdroje nahrávok. Ak používateľ interaguje s externým obsahom, napríklad so sociálnymi sieťami alebo webovou stránkou, a zadá túto adresu do prehliadača, systém ho automaticky presmeruje do nového okna alebo karty v prehliadači.

Každá kategória výberu môže mať pridelený špecifický event, ktorý bude spracovávať všetky akcie danej kategórie. Tento mechanizmus umožňuje vysokú mieru prispôbitelnosti pri aplikovaní rôznych šablón a poskytuje možnosť definovať individuálne správanie pre každú vlastnosť.

Cieľom návrhu bolo odstrániť komplikované dedičné vzťahy a zvýšiť flexibi-



Obr. 6.4: Ukážka nastaviteľného umiestnenia okna UI menu pre potreby analýzy, spolu s možnosťou listovania výberu scény. (zdroj: NDMVR)

litu interakcie s menu v rámci aplikácie. Zmena vnútorného stavu šablóny (template) sa zaznamenáva a spracováva okamžite, bez potreby nepretržitého renderovania v pozadí frameworku (napr. React). Tento prístup bol realizovaný pomocou npm balíčka ², ktorý umožňuje efektívnu komunikáciu medzi rôznymi časťami aplikácie prostredníctvom eventového systému.

Každá poskytnutá funkcia je priradená ku konkrétnemu eventu a odoberatelia (subscribers) môžu na tieto udalosti reagovať. Každá inštancia onclick komponentu v rámci šablóny prenáša informácie o vykonanej akcii vrátane potrebných dát, názvu akcie a identifikátora zvoleného objektu. Schéma umožňuje nastaviť eventy pre jednotlivé vlastnosti (properties), čím poskytuje možnosť zachytávať akcie na viacerých úrovniach – od špecifických properties až po globálne nastavenia celého systému. Tento prístup ponúka vysokú mieru flexibility a umožňuje optimalizáciu použitia podľa konkrétnych potrieb implementácie.

Tieto zmeny boli aplikované do všetkých definovaných šablón uvedených v časti 4.2.5.

Ďalším z prvkov bola možnosť zmeny polohy UI menu, teda jeho posunutie v priestore a následné zapamätanie pozície, podobne ako to ponúkajú aj iné používateľské rozhrania VR headsetov a ich platforiem. Táto možnosť je nezávislá od priestoru a je zobrazená na obr. 6.4.

Jej pozíciu zabezpečila implementácia Three.js, uvedená v časti 1.2, v kombinácii s knižnicou³, ktorá poskytla kinematiku použitú pre okno kontextu.

²<https://www.npmjs.com/package/fbemitter>

³<https://github.com/binzume/aframe-xylayout>

6.2 Optimalizácia vizuálnej prezentácie scény

Navrhované riešenie vychádza z nedostatkov identifikovaných respondentmi, ktorí poukázali na vizuál, ktorý nebol úplne aktuálny a uspokojivý, ako je opísané v časti 5.3.7. Napriek už spomenutým obmedzeniam samotnej renderovacej technológie sme sa rozhodli premietnuť určité pripomienky respondentov do návrhu novej scény.

6.2.1 Návrh vizuálnej úpravy scény

Táto scéna sa bude niesť v duchu jednoduchosti, keďže rôzne farebné prvky často nepôsobili dostatočne profesionálne a nevyvolávali požadovaný dojem. Usilujeme sa preto o jednotný vizuálny štýl, ktorý bude implementovaný do jednoduchšieho modelu či prostredia. Scéna zostane vizuálne zaujímavá, no nebude obsahovať okolité kopce, stromy ani ďalšie rušivé prvky. Jej vzhľad by mal aspoň čiastočne pripomínať vnútornú štruktúru detektora, čím podporí lepšiu angažovanosť v danej analytickej sfére.

Jednou z hlavných požiadaviek na dizajn bolo vytvorenie vizuálneho štýlu, ktorý evokuje futuristický vzhľad a nadčasovosť riešenia. Mnohí respondenti vyjadrili záujem o prvky, ako je lietanie či opätovný pád na zem, teda prvky spojené s adrenalínom a intenzívnymi zmyslovými vnemami v danej scéne. Na základe tejto spätnej väzby sme sa rozhodli implementovať podobné prvky aj v rámci nášho návrhu. Veríme, že takáto dynamická vizuálna skúsenosť zvýši angažovanosť používateľov a prinesie im väčšiu radosť z interakcie so scénou, pričom ju nebudú vnímať iba ako analytický nástroj.

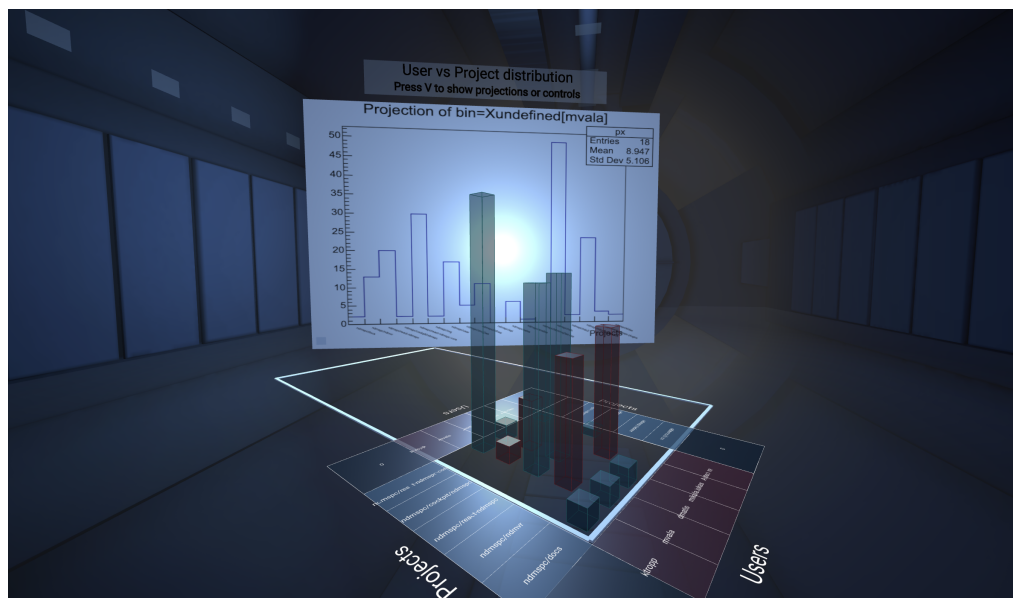
6.2.2 Implementácia vizuálnej úpravy scény

Scéna je definovaná ako samostatná React komponenta, ktorá sa postupne renderuje po jej výbere prostredníctvom UI menu. Jej základ tvorí voľne dostupný model ponúkaný ako príklad práce s frameworkom A-Frame ⁴.

Základ renderovania plochy histogramu bol upravený tak, aby si zachoval dostatočnú veľkosť aj pri zväčšujúcom sa počte binov. Plocha histogramu je renderovaná s offsetom, ktorý sme vyhodnotili ako viac než postačujúci. Celú scénu obklopuje model, pričom hlavná analytická časť – kino, histogram a jeho osi – sa nachádza v jeho strede.

⁴<https://aframe.io/aframe/examples/showcase/anime-UI/models/engine.glb>

Podlaha je zámerne priehľadná, čím vytvára dojem, že používateľ interaguje s histogramom na sklenenej podlahe a vníma pod sebou určitú hĺbku. Pre niektorých používateľov môže byť tento efekt nepríjemný, preto sme ponechali aj klasické zobrazenia ponúkané v rámci NDMVR [7]. V takýchto prípadoch odporúčame ich využitie.



Obr. 6.5: Tmavé zobrazenie s podporou bodového svetla (zdroj: NDMVR)

Scéna je vizuálne pomerne jednoduchá a jednofarebná. Obsahuje imitácie svetiel umiestnených po stranách a v strede modelu, čím sa podporuje laboratórna atmosféra. Používateľ si môže prostredníctvom UI menu vybrať rôzne svetelné režimy – od svetlej verzie až po postupné tlmenie osvetlenia, pričom najtmavšia verzia (Darker) zachováva iba osvetlenie plátna. Jeden z režimov osvetlenia je uvedený na obr. 6.5.

V časti 6.3 sa venujeme úprave vizualizácie binov a ich vylepšeniam. V kombinácii s tmavou tematikou osvetlenia sa odporúča použitie svietiacich binov a osí, ktoré sú ideálne pre tmavší svetelný režim. Celková funkcionálna pôvodných scén bola zachovaná v závislosti od plátna a histogramu.

6.3 Modifikácia vizualizácie histogramu

Po zlepšení vizuálu scény sme sa zamerali na spôsob zobrazenia histogramu, konkrétne jeho osí a binov. Na základe výsledkov z časti 5.3.7, kde poukazujeme na nedostatky v zobrazovaní binov — ktoré doteraz predstavovali iba jednoduché objektové kocky s výraznými farebnými plochami slúžiacimi ako osi — sme sa rozhodli tieto nedostatky odstrániť a vizuálne ich vylepšiť.

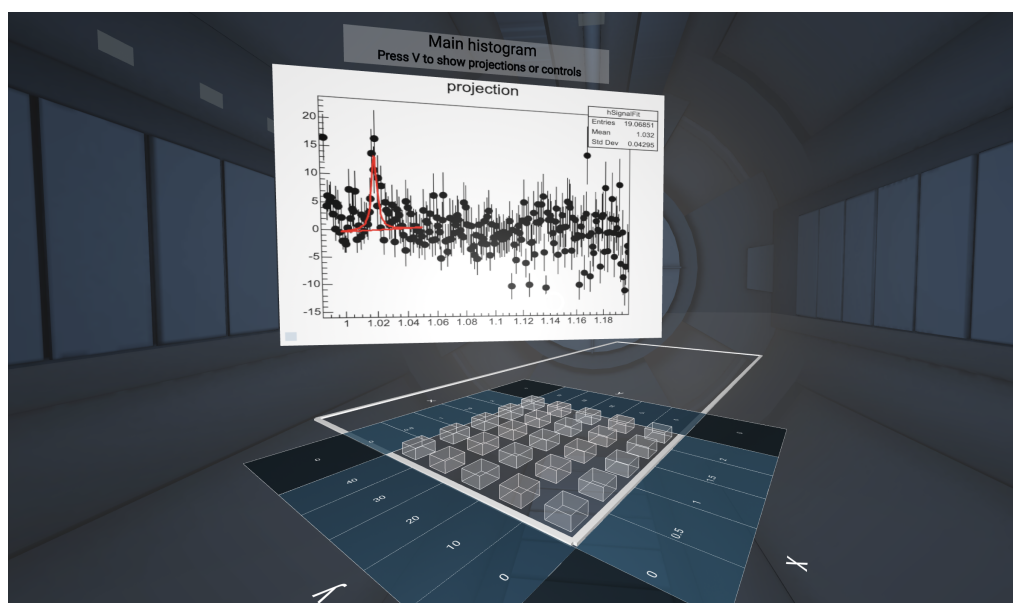
6.3.1 Návrh prispôsobenia dizajnu histogramu

Vzhľadom na časové obmedzenia a rozsah už implementovaného riešenia renderovania binov, ako aj na náročnosť hľadania vhodnej alternatívy, nebolo možné zásadne meniť ich základnú štruktúru. Navyše, keďže steny kociek budú v budúcnosti slúžiť na zobrazovanie rôznych foriem obsahu, úprava ich geometrie nepripadala do úvahy.

Snažili sme sa však podporiť futuristický dizajn aj prostredníctvom binov a vizuálne ich zladíť s jednotnou farebnou schémou scény uvedenej v časti 6.2. Na tento účel využijeme JSON definície tém poskytované NMDVR, pričom bude potrebné vykonať niekoľko úprav na podporu požadovanej funkcionality.

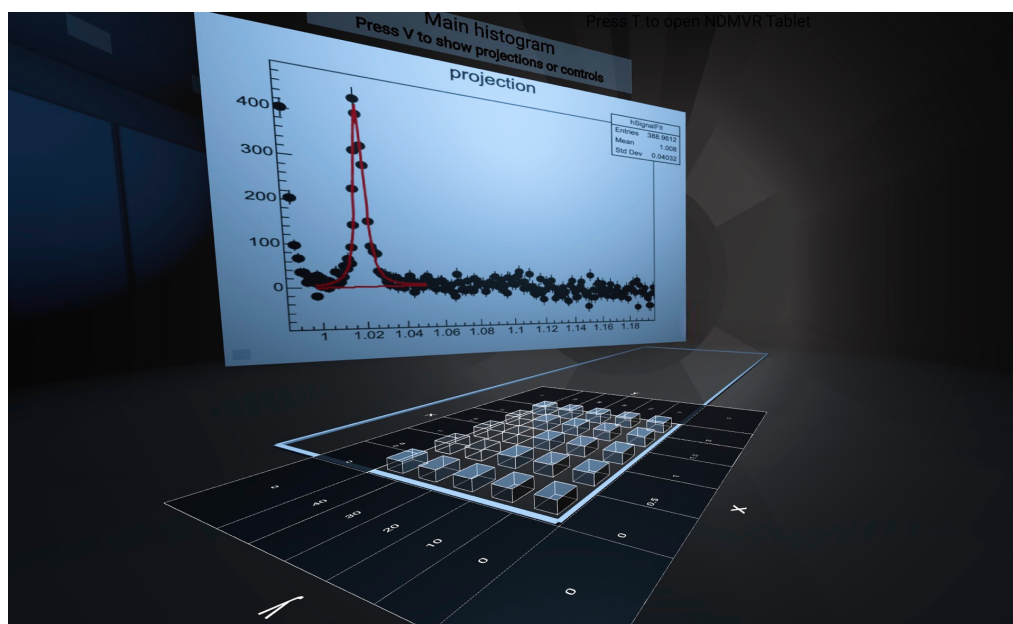
6.3.2 Vizualizácia histogramu

V rámci vylepšení vizualizácie histogramu sme do komponentu ThemeProvider pridali podporu pre nastavenie nepriehľadnosti (opacity) pre každý bin, ako aj možnosť dynamického renderovania binov v rôznych farebných paletách. Okrem toho sme upravili aj farby osí a farby primárnych a sekundárnych textových prvkov. Výsledné zobrazenie je uvedené na obr. 6.6.



Obr. 6.6: Aktualizovaný vizuál histogramu v svetlom režime scény (zdroj: NDMVR)

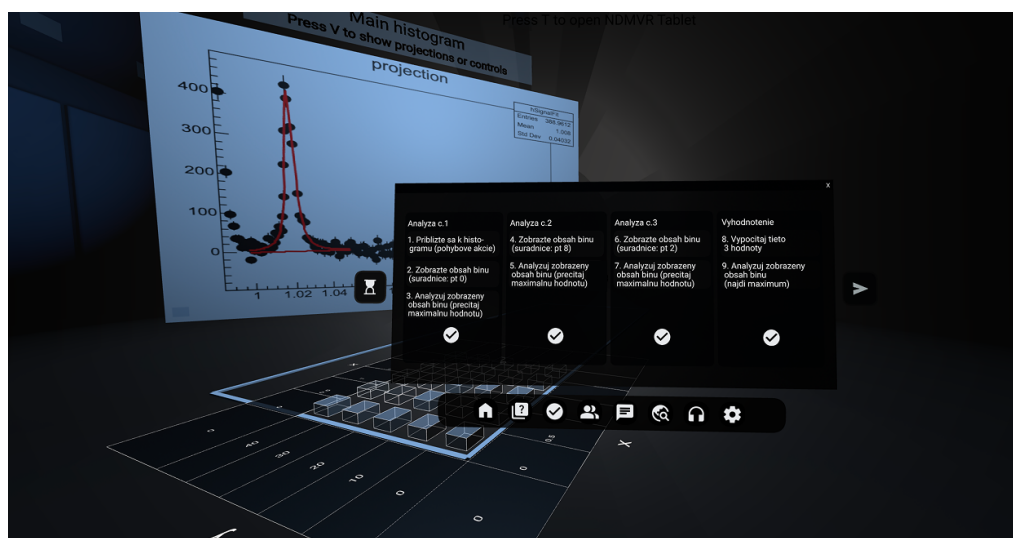
Na podporu moderného a futuristického vizuálu sme do každého A-Frame primitívu (boxu) reprezentujúceho bin pridali nový A-Frame komponent. Tento komponent zohľadňuje rozmery binu, ktoré sa prispôbujú na základe jeho nastavení a veľkosti. Parametre sa dynamicky aktualizujú, podobne ako v prostredí



Obr. 6.7: NDMVR scéna: Nastavenie tmavšieho osvetlenia s úpravou binov a osí pre svetelné orámovanie(zdroj: NDMVR)

React, a určujú ohraničenie binu výpočtom a vykreslením línií v rámci komponentu obsahujúceho grafické primitíva. Línie sa zobrazujú v súlade so zvolenou farebnou témou a vstupnými parametrami, ktoré ovplyvňujú výpočet jednotlivých hrán.

Do budúcnosti bude možné tento komponent využiť aj na vizualizáciu zmien v histograme, ako aj na zobrazenie len obrysov v porovnaní s pôvodnou alebo maximálnou veľkosťou binu. Okrem toho bude možné zobrazíť dva biny súčasne a vizuálne ich odlišiť.



Obr. 6.8: Zobrazenie UI menu pri minimálnom osvetlení novej scény (zdroj: NDMVR)

Vo väčšine predvolených tém sú vizuálne zmeny minimálne. Pri upravených témach je však hodnota nepriehľadnosti (opacity) znížená, čo umožňuje zvýrazniť len obrys (kostru) binu so slabším obsahom. V niektorých témach rám binu „svieti“ pri tmavších zobrazeniach, čím sa dosahuje moderný a futuristický efekt, ktorý v prostredí NDMVR predtým nebol dostupný. Ukážka tohto riešenia je znázornená na obr. 6.7.

Na obrázku 6.8 je znázornené využitie nového vizuálneho štýlu v kombinácii s UI menu, ktoré lepšie zapadá do celkového prostredia a konceptu novej vizualizácie. Funkcionalita používateľského rozhrania zostala zachovaná a nebola ovplyvnená implementáciou týchto vizuálnych úprav.

7 Vyhodnotenie

V tejto časti prezentujeme hodnotenie navrhnutého riešenia na základe údajov získaných počas hromadného testovania komponentu NDMVR, opísaného v kapitole 5. Hodnotenie vychádza z kombinácie objektívnych a subjektívnych údajov, analyzovaných v časti 2.3, ktoré boli zhromaždené počas testovania. Cieľom je zhodnotiť celkový vplyv implementovaných zmien na výsledky testovania a identifikovať potenciál pre ďalšie zlepšenia vyplývajúce z realizovaných úprav systému.

7.1 Zhrnutie priebehu hodnotenia NDMVR

V kapitole 5 venovanej hodnoteniu používateľskej skúsenosti bolo testovanie realizované podľa vopred definovaného scenára, metodík, očakávaného prínosu ich výsledkov. Testovací scenár pozostával zo štyroch častí, ktoré zahŕňali rôzne úlohy reflektujúce reálne používanie systému. Výber respondentov bol ovplyvnený konaním výročnej udalosti CERN-u, počas ktorej testovanie prebiehalo. Výber zúčastnených osôb vhodne zodpovedal cieľovej skupine používateľov – išlo prevažne o zamestnancov CERN-u, spolupracovníkov, pedagógov Katedry jadrovej fyziky Univerzity Pavla Jozefa Šafárika, ako aj o širšiu verejnosť so záujmom o problematiku jadrového výskumu.

Primárne hodnotiace dáta boli zbierané prostredníctvom sumatívnej metódy hodnotenia používateľskej skúsenosti (UES), pričom bola použitá dotazníková metóda prispôbena pre VR prostredie a implementovaná priamo v používateľskom rozhraní NDMVR menu. Metodika UES sa zameriavala na štyri základné faktory opísané v časti 2.5.2, ktoré predstavujú subjektívne hodnotiace premenné.

Na zvýšenie presnosti výsledkov boli v súlade s odporúčaniami podľa analyzovanej časti 2.3 zbierané aj objektívne dáta, konkrétne čas vykonávania jednotlivých úloh v rámci testovacieho scenára. Na doplnenie kvalitatívnych údajov bol každý respondent po absolvovaní testovania vyzvaný na krátky rozhovor, v ktorom mohol vyjadriť svoj subjektívny dojem z používania systému. Počas testovania

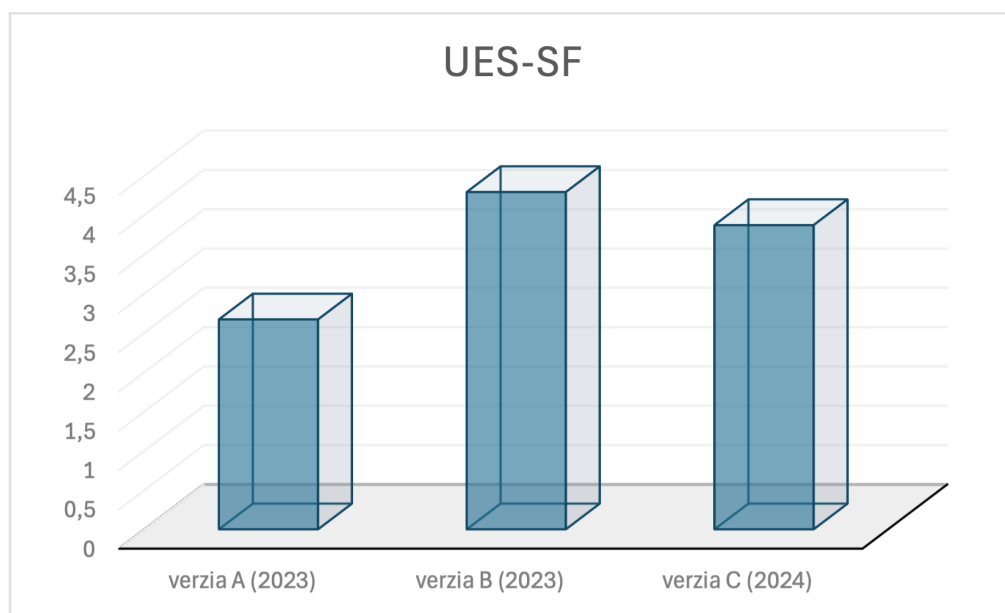
vania boli zároveň zaznamenávané pripomienky a poznámky používateľov uvedené v časti 5.3.7, čo poskytlo dôležité kvalitatívne poznatky na identifikáciu a následné odstránenie nedostatkov.

Kombinácia získaných kvantitatívnych a kvalitatívnych údajov poskytla hodnotný základ pre návrh vylepšení používateľského rozhrania, ako aj celkového UX komponentu NDMVR. Testovanie zároveň poslúžilo ako spätná väzba k súčasnemu stavu riešenia a identifikovalo priestor na jeho ďalší rozvoj.

7.2 Retrospektívne porovnanie UX hodnotenia

Pri zhrnutí výsledkov sumatívneho charakteru sme vychádzali z práce [7], ktorá v roku 2023 realizovala meranie používateľskej skúsenosti komponentu NDMVR. Súčasťou tohto merania bola okrem iného aj aplikácia metodiky UES, ktorej výsledky sme sa rozhodli použiť na účely porovnania s testovaním, ktoré je obsiahnuté v tejto práci.

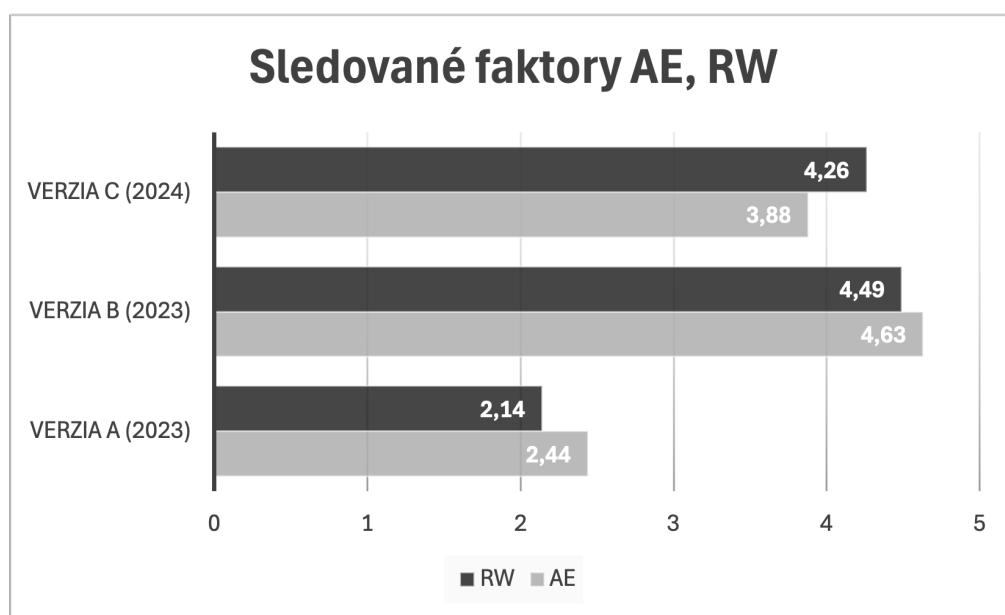
V práci [7] boli merané dve verzie: pôvodná verzia komponentu NDMVR a neskôr verzia upravená na základe zistení a odstránených nedostatkov identifikovaných počas testovania v roku 2023. Cieľom tohto porovnania je zdokumentovať postupný vývoj systému a zároveň poskytnúť presnejší pohľad na jeho funkčnosť vzhľadom na počet respondentov, ich odborné zázemie, mieru záujmu a potenciálne možnosti využitia.



Obr. 7.1: Priemerná angažovanosť vo verziách NDMVR podľa UX testovania (UES-SF)

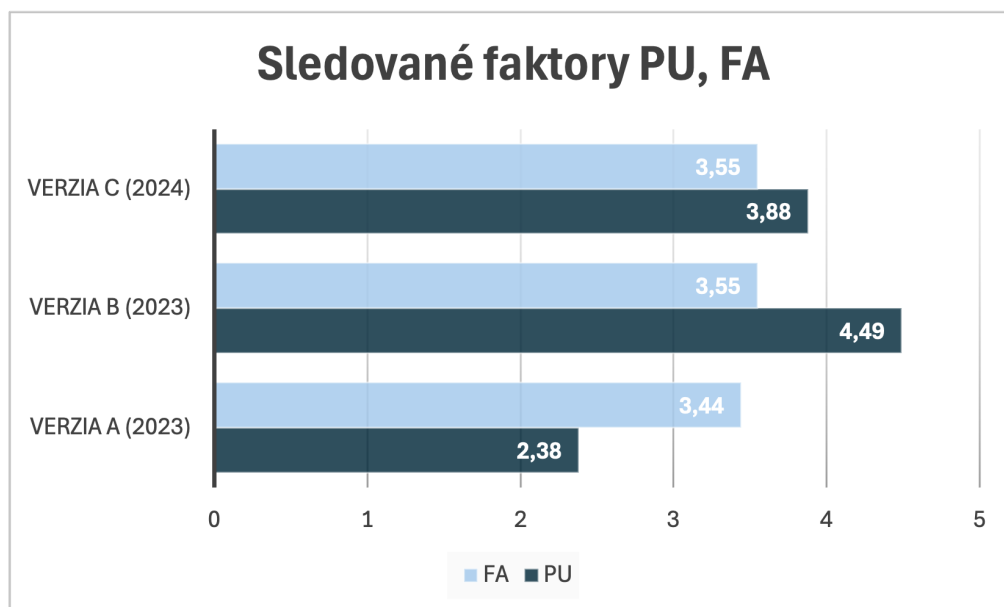
Výsledky zobrazené na obr. 2.5.2 sledujú štyri faktory UES (User Experience Satisfaction), konkrétne estetickú príťažlivosť (AE – Aesthetic Appeal), vnímanú použiteľnosť (PU – Perceived Usability), faktor odmeny (RW – Reward Factor) a sústredenú pozornosť (FA – Focused Attention) naprieč 3 verziami NDMVR a 3 meraniami. Meranie z roku 2024 vykazuje v tomto prípade **FA = 3,55, AE = 3,88, PU = 3,88 a RW = 4,26**. Priemerné skóre angažovanosti je **3,87** na škále UES od 1 (najhoršie hodnotenie) do 5 (najlepšie hodnotenie), čo môžeme hodnotiť ako **dobré**.

Výsledky z pôvodnej práce [7] považujeme za menej presné, a to z dôvodu odlišného spôsobu zberu dát. Používateľský zážitok bol počas testovania opakovane prerušovaný a dáta boli zaznamenávané mimo kontextu interakcie. V testovacej metodike taktiež chýbalo digitálne zobrazenie scenára a nebolo zabezpečené časové sledovanie úloh, čo negatívne ovplyvnilo presnosť objektívneho vyhodnotenia jednotlivých častí. Uvedené rozdiely sú podrobnejšie opísané v časti 5.1.



Obr. 7.2: Priemerná estetická príťažlivosť spolu s faktorom odmeny pri používaní rôznych verzií NDMVR v závislosti od UX testovania (UES-AE, UES-RW)

Výsledky meraní faktorov AE a RW sú znázornené na obr. 7.2, zatiaľ čo výsledky meraní faktorov PU a FA sú zobrazené na obr. 7.3. Za pokles v porovnaní s verziou z roku 2023 mohla presnejšia forma testovania, vyšší počet respondentov a lepšie optimalizovaná testovacia scéna, ktorá bola vytvorená špeciálne na účely hodnotenia. Zároveň však išlo o jednoduchšie spracované prostredie, ktoré mohlo pôsobiť menej pôsobivo. Aj samotná prezentácia riešenia bola v tejto verzii mierne utlmená, keďže jej cieľom bolo jasnejšie sprostredkovať konkrétny účel



Obr. 7.3: Priemerná vnímaná použiteľnosť spolu so sústredenou pozornosťou pri používaní rôznych verzií NDMVR v závislosti od UX testovania (UES-PU, UES-FA)

systému a jeho pochopenie.

V neposlednom rade sa na výsledkoch negatívne prejavil aj fakt, že tablet ako súčasť rozhrania bol v testovanej verzii len minimálne rozpracovaný a prakticky nefunkčný, čo viacerí respondenti aj výslovne spomenuli vo svojich spätných väzbách.

Na základe aplikovanej metodiky a zvoleného prístupu v našom testovaní môžeme konštatovať, že nami realizované testovanie prinieslo presnejšie výsledky vo všetkých sledovaných faktoroch. Výsledky tak poskytnú dôveryhodnejšie hodnotenie používateľskej skúsenosti (UX) komponentu NDMVR a zároveň poukázali na prínosy implementovaných zmien, ako aj potenciál pre ďalšie zlepšenia.

7.3 Vyhodnotenie priebehu vykonávania úloh a správania respondentov

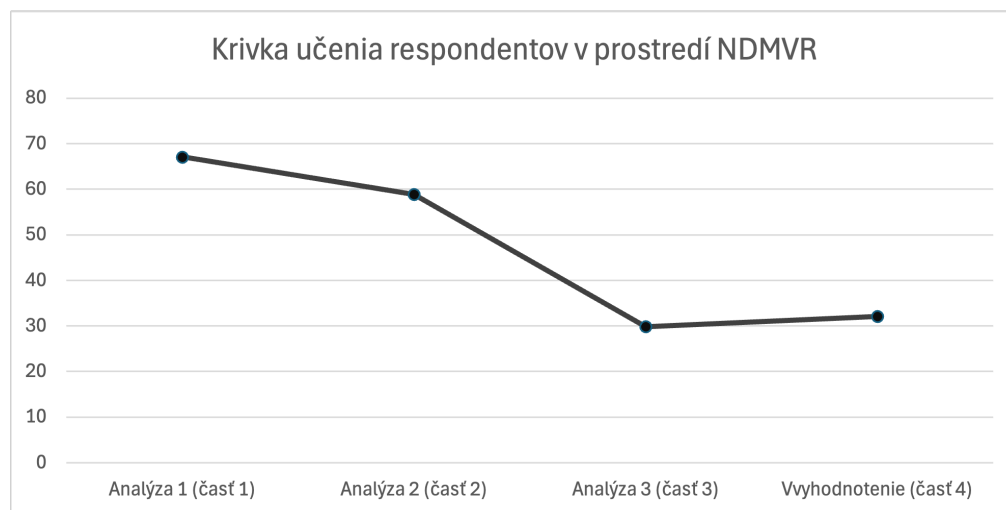
Sledovanie vykonávania úloh prebiehalo podľa štyroch definovaných častí testovacieho scenára uvedeného v časti 5.1. Výsledky a správanie respondentov sú detailnejšie popísané v časti 5.3.7. Zaznamenané boli merateľné časové údaje, pozorovania počas interakcie, ako aj reakcie účastníkov po skončení testu.

7.3.1 Časová náročnosť úloh

Priemerné časy pre jednotlivé časti scenára boli nasledovné:

- **Časť 1 (Inicializácia analýzy a zoznámenie sa s histogramom):** 67 sekúnd
- **Časť 2 (Základná analýza a interpretácia dát):** 58 sekúnd
- **Časť 3 (Opakovaná analýza dát):** 29 sekúnd
- **Časť 4 (Vyhodnotenie údajov):** 32 sekúnd

Celkový priemerný čas na dokončenie celého scenára bol približne **3 minúty**. Postupný pokles v čase vykonávania poukazuje na efekt znižujúcej sa krivky učenia, ktorá je zobrazená na obr. 7.4. Používatelia si postupne osvojili ovládanie a interakciu s prostredím NDMVR, čo viedlo k efektívnejšiemu riešeniu úloh. Bližšie vyhodnotenie tejto sekcie testovania je obsiahnuté v časti 5.3.6.



Obr. 7.4: Krivka osvojovania ovládania a orientácie v NDMVR

7.3.2 Pozorovania a problémy počas testovania

Počas testovania boli zaznamenané rôzne relevantné pozorovania, ktoré významne prispeli k hlbšiemu porozumeniu správania používateľov a interakcii s testovaným systémom. Už v prvej časti scenára sa u dvoch respondentov objavila výraznejšia miera neistoty a chybovosti, čo možno pripísať ich prvej interakcii s VR prostredím a ovládacími prvkami.

Druhá časť testovania bola z pohľadu používateľského pochopenia najstabilnejšia – všetci respondenti ju zvládli bez problémov, pričom jej obsah a forma boli vnímané ako zrozumiteľné a intuitívne.

V tretej časti, ktorá si vyžadovala opakovanie už vykonanej analýzy, jeden z účastníkov testovanie nedokončil. Dôvodom však nebola nepochopiteľnosť úloh, ale externé faktory – konkrétne časová tieseň a iné povinnosti. Respondent svoju skúsenosť hodnotil pozitívne a vyjadril ľútosť, že test nemohol dokončiť.

Najväčší výskyt chýb sa objavil v štvrtej časti testovania, zameranej na vyhodnotenie dát. Napriek tomu, že tlačidlo na spustenie vyhodnotenia bolo vizuálne zvýraznené, až piati respondenti s jeho použitím zápasili. Niektorí z nich potrebovali krátku asistenciu, iní sa rozhodli skúsiť údaje analyzovať manuálne – bez pomoci systému. Možné príčiny tohto správania sú viaceré: jednak mohlo ísť o stres z prebiehajúcej skúsenosti alebo zníženú sústredenosť, no zároveň sa ukázalo, že nie všetky funkcionality systému boli dostatočne objasnené pred začiatkom úloh. Navyše, pre väčšinu respondentov predstavovalo prostredie NDMVR nový koncept interaktívnej analýzy dát, čo mohlo prispieť k neistote pri ovládaní. Tomuto faktoru zodpovedá aj mierny nárast časovej náročnosti v časti 4 (vyhodnotenie) scenára.

Ďalším faktorom ovplyvňujúcim orientáciu bola samotná povaha 360-stupňového VR prostredia. Niektorí respondenti sa nedokázali správne natočiť a zamerať svoj výhľad smerom k prvkom rozhrania – hoci boli zobrazené v ich blízkosti. Tento problém sa výraznejšie prejavoval u používateľov bez predchádzajúcej skúsenosti s VR technológiou, najmä počas dlhšej interakcie. S rastúcim časom sa v niektorých prípadoch zvyšovala aj únava a znížená schopnosť orientácie v priestore.

7.3.3 Subjektívne hodnotenie a spätná väzba

Väčšina respondentov hodnotila zážitok ako pozitívny. Oceňovali najmä rýchlosť analýzy a výsledkov, pričom opakovane zaznela reakcia v duchu: *„Očakával som, že to bude zložitejšie, no bolo to prekvapivo jednoduché.“*

Po počiatočnom zoznamovaní sa so spôsobom interakcie – najmä ovládaním pomocou Oculus kontrolerov – sa ovládanie javilo ako intuitívne a úlohy zvládnutelné. U respondentov s predchádzajúcou skúsenosťou s VR bol priebeh testovania plynulejší a časová záťaž v jednotlivých častiach výrazne nižšia už od začiatku.

Výsledky boli doplnené aj o dotazník ¹, ktorý respondenti dobrovoľne vyplnili počas testovania. Obsahoval základné informácie o účastníkoch, ktoré umožnili presnejšie vyhodnotenie skúseností s VR, používateľských preferencií a kvalitatívnych poznatkov.

¹<https://forms.gle/MV7CQcqEsyH4ss5w6>

7.4 Riešenie zistených nedostatkov: Prijaté opatrenia

Spätná väzba získaná ústnou formou od respondentov odhalila viacero nedostatkov, ktoré sú bližšie opísané v časti 5.3.7. Keďže išlo o vzorku používateľov s rôznorodým odborným zázemím, v ktorej väčšinu tvorili zdatní odborníci z fyzikálneho prostredia (zamestnanci CERN-u, pedagógovia a odborníci na dátovú analýzu), túto skupinu považujeme za dôveryhodnú a vysoko relevantnú z hľadiska hodnotenia komponentu NDMVR. Avšak, je potrebné poznamenať, že medzi respondentmi sa nachádzali aj niektorí účastníci s nižším vzdelaním alebo bez vysokoškolského stupňa, ako napríklad študenti, čo môže do istej miery ovplyvniť interpretáciu výsledkov.

Napriek tomu výpovede odborníkov, ako aj týchto menej skúsených respondentov, priniesli cenné a relevantné informácie. Kombinácia ich rôznych perspektív umožnila vytvoriť vyvážený obraz o silných a slabých stránkach komponentu NDMVR. Práve tieto výpovede sa stali základom pre mnohé z vykonaných zmien, ktoré sú podrobnejšie rozpracované v nasledujúcej časti 5.3.7.

V nasledujúcej časti sú zhrnuté hlavné oblasti, na ktoré sa úpravy zameriavali:

7.4.1 Nedostatočná podpora používateľského rozhrania

Ako je uvedené v časti 3, komponent NDMVR a framework A-frame dlhodobo nedisponovali robustným systémom pre tvorbu znovupoužiteľných, konzistentných a vizuálne príťažlivých používateľských prvkov vo VR prostredí. UI komponenty boli spravidla založené na manuálne vytváraných primitívach (napr. jednoduché boxy s aplikovaným materiálom), ktoré len obmedzene pokrývali požiadavky na kvalitný používateľský zážitok. Ich vizuálna atraktivita bola nízka, funkcionálna často obmedzená a celková použiteľnosť nedostatočná, čo sa prejavilo aj v testovanej verzii používateľského „tabletu“ (pozri 4.2).

Na základe analýzy súčasného stavu, identifikovaných potrieb a porovnania dostupných riešení od viacerých vývojárov pre A-Frame (s dôrazom na zachovanie minimálnych kritérií kvalitného UX v rôznych typoch VR aplikácií, ktoré sú analyzované v časti 2.2.3 a neskôr premietnuté do návrhu v časti 4.2.2), bola v tejto práci navrhnutá a implementovaná modulárna UI knižnica pre A-Frame. Táto knižnica predstavuje jeden z kľúčových prínosov diplomovej práce.

Navrhnutá knižnica umožňuje jednoduchú, flexibilnú a opakovane použiteľnú integráciu používateľských prvkov – nielen do centrálného „menu“, ale aj kdekoľvek v rámci 3D prostredia. Zároveň počíta s možnosťou rozširovania a udržiavateľnosti, čím výrazne uľahčuje budúci vývoj rozhraní v rámci komponentu

NDMVR a ďalších projektov využívajúcich A-frame.

Spätná väzba od respondentov

Respondenti počas testovania vyjadrili spokojnosť s intuitívnosťou menu, ktoré využíva prvky tejto knižnice, ako aj s jeho dizajnom. Viacnásobne bolo pozitívne hodnotené, že menu zostáva pripevnené ku kamere, čím sa výrazne zjednodušila orientácia v priestore a zároveň eliminovala potreba dodatočného hľadania ovládacích prvkov.

Viacerí respondenti vnímali ako pozitívne, že ovládacie prvky boli konzistentné a vizuálne podobné, čo prispelo k rýchlejšiemu osvojeniu si práce so systémom.

Niekoľko používateľov uviedlo problém s čitateľnosťou textu kvôli nenoseným okuliárom vo VR headsete. Avšak po odporúčaní priblížiť sa k menu (ktoré ostáva v priestore fixované), sa čitateľnosť výrazne zlepšila a tento aspekt bol následne hodnotený ako veľmi prínosný.

Menu sa stalo najčastejšie využívaným a skúmaným prvkom – nie len ako nástroj, ale aj ako spôsob, akým sa používatelia intuitívne orientovali v ponúkaných možnostiach. To svedčí o vysokej použiteľnosti aj pre menej technicky zdatných používateľov a naznačuje, že navrhované zmeny smerovali správnym smerom.

7.4.2 Vylepšenie vizuálnej konzistencie a používateľskej interakcie v NDMVR

Ďalším významným nedostatkom predchádzajúcich verzií komponentu NDMVR [7], ako aj aktuálne analyzovanej verzie v kapitole 3, bola absencia esteticky príťažlivých a konzistentných UX prvkov v rámci celého systému. Rôzne časti komponentu alebo jednotlivé UI prvky síce plnili svoju funkciu, no často pôsobili dojmom provizórnych riešení – išlo prevažne o samostatne vložené objekty, ktoré neboli prepojené s jednotným dizajnovým štýlom systému a nezabezpečovali minimálnu úroveň kvalitného UX.

Ako reakcia na tieto zistenia bola v rámci práce navrhnutá štandardizácia vizuálu a správania interaktívnych prvkov prostredníctvom opakovane použiteľných šablón a ich základných prvkov (napr. tlačidlá, layout boxy, informačné panely, content okná). Cieľom týchto úprav nebolo len vizuálne vylepšenie rozhrania, ale predovšetkým zabezpečenie konzistentnej interakcie naprieč celým systémom, aby používateľ nemusel pri každom prvku nanovo zisťovať spôsob jeho ovládania. V prípade rôznych budúcich implementácií je v práci poskytnutá

možnosť preddefinovaného testovania UX, ktoré ponúka UI menu pre rýchlu analýzu rôznych faktorov podľa UES metodiky alebo čistej použiteľnosti prostredníctvom SUS metodiky. Toto testovanie je k dispozícii bez akejkoľvek predchádzajúcej prípravy a umožňuje zber údajov, ktoré môžu byť následne vyhodnotené s cieľom poskytnúť spätnú väzbu na otázky týkajúce sa UX a jeho konzistencie v rámci systému pri narastajúcom obsahu komponentov.

7.4.3 Nedostatočná funkcionálna a vizuálna spracovanie tabletu

Jedným z najčastejšie kritizovaných prvkov podľa časti 5.3.7 bol tzv. tablet, ktorého funkcia a prevedenie podľa respondentov výrazne znižovali komfort pri práci s komponentom NDMVR. Bol neprakticky umiestnený na ľavej ruke a ovládaný pravou rukou, čo viacerí označili ako neintuitívne a zavadzajúce. Navyše jeho vizuálny dizajn nezapadal do celkového prostredia, čo u používateľov vyvolávalo frustráciu a rušilo plynulosť zážitku.

Na základe týchto výpovedí bola jeho funkcionálna presunutá do hlavného UI menu. Nové riešenie ponúka nielen elegantnejší vzhľad, ale aj lepšiu prístupnosť a intuitívnejšie ovládanie. Podrobnosti o tejto zmene sú uvedené v časti 6.1.

7.4.4 Vylepšenie vizuálneho realizmu a modernizácia dizajnových prvkov

Viacerí respondenti v časti 5.3.7 poukázali na nedostatočný vizuálny realizmus prostredia, čo negatívne ovplyvňovalo pocit profesionality. Dizajn niektorých prvkov bol vnímaný ako zastaraný, miestami až detinský. Hoci zmena samotnej VR technológie nebola možná v rámci tejto práce, boli prijaté kroky na zlepšenie vizuálnej kvality a modernejšieho vzhľadu kľúčových nástrojov, ako sú histogram a zobrazenie binov.

Zachovali sme pôvodnú štruktúru a intuitívnosť vizualizácie, no dizajn bol vylepšený o osvetlené hrany, efekt prechodov, priehľadnosti a futuristickejšie obrysy binov. Histogram bol doplnený o zvýraznenie hrán a lepšiu čitateľnosť v tmavších scénach. Výsledné zmeny sú bližšie ilustrované v časti 6.3.

7.4.5 Nová scéna v prostredí

Pôvodné VR scény, vytvorené v práci [7], boli ponechané kvôli ich univerzálnosti a možnosti ďalšej úpravy. Napriek tomu sme na základe spätnej väzby uvedenej v časti 5.3.7 implementovali novú scénu, ktorá evokuje prostredie laboratória či

detektora. Táto scéna zahŕňa sklenenú podlahu, rôzne úrovne osvetlenia a dodáva jemný prvok adrenalínu, ktorý bol podľa výpovedí respondentov vítaným spestrením.

Nové prostredie taktiež ponúka možnosti prispôsobenia vizuálnych parametrov osvetlenia, čo zvyšuje komfort pri dlhodobom používaní. Zameranie ostáva na jednoduchosť, kontrast a podporu efektívnej práce s dátami.

7.4.6 Vylepšené ovládanie pohybu

Ako je popísané v časti 4.1, boli vykonané významné úpravy v oblasti tzv. fly módu, ako aj pohybu v desktopovom a VR zobrazení. Pohyb bol implementovaný plynulejšie vzhľadom na scénu, čím sa výrazne zlepšila navigácia a znížila miera dezorientácie, najmä pri práci mimo VR prostredia. Tieto zmeny vznikli z iniciatívy autora a prispievateľov komponentu NDMVR, ako aj na základe analýzy vykonanej ešte pred UX testovaním.

Ako uvádzame v časti 5.3.7, respondenti počas testovania nevykazovali známky problémov v rámci ovládania pohybu ani nepríjemných fyziologických reakcií (napr. nevoľnosť). Naopak, niektorí z nich explicitne vyjadrili nadšenie z nového fly módu, ktorý podľa ich slov ponúkol adrenalínový zážitok a zároveň intuitívne reagoval na smer pohľadu a ovládačov.

7.5 Zhrnutie vyhodnotenia

Vyhodnotenie testovania naznačuje, že vykonané zmeny mali pozitívny vplyv, najmä v súvislosti s vylepšenými prvkami (menu, pohyb), a prispeli k zlepšeniu presnosti UX hodnotenia komponentu NDMVR, pričom odhalili nové oblasti vyžadujúce ďalšie zlepšenie. Hoci z dôvodu obmedzeného času a organizačných príčin nebolo možné realizovať rozsiahle opätovné testovanie, ktoré by kvantitatívne potvrdilo tieto zlepšenia, spätná väzba poskytuje silné dôkazy o nedostatkoch komponentu NDMVR, ktoré sú podrobne popísané v časti 6. Predpokladáme, že odstránenie týchto nedostatkov povedie k celkovému zlepšeniu UX. Pre ďalší vývoj a vylepšenie systému je možné vykonať ďalšie testovania, ktoré by umožnili podrobnejšie vyhodnotenie implementovaných zmien, identifikovali ďalšie oblasti na zlepšenie a potvrdili úspešnosť vykonaných úprav na základe metodiky testovania popísanej v kapitole 5 a poskytnutého rozhrania „menu“.

8 Záver

Cieľom diplomovej práce bolo analyzovať súčasný stav webového komponentu NDMVR, určeného na vizualizáciu údajov z fyzikálnych experimentov vo virtuálnej realite, so zameraním na jeho konfigurovateľnosť, používateľský zážitok a funkčnosť používateľského rozhrania. Na základe tejto analýzy bola vytvorená prvá verzia upraveného komponentu, ktorá bola následne testovaná s cieľom identifikovať jej silné a slabé stránky z pohľadu používateľskej skúsenosti. Získané poznatky boli využité pri návrhu a implementácii konkrétnych vylepšení, ktoré vznikli v spolupráci s ďalšími riešiteľmi projektu NDMVR.

V rámci práce prebehlo testovanie prostredia s využitím navrhnutého používateľského rozhrania na reprezentatívnej vzorke 25 respondentov. Výsledky testovania identifikovali konkrétne oblasti, ktoré ovplyvňovali používateľskú spokojnosť, orientáciu a interakciu v prostredí VR. Na základe týchto zistení boli vykonané úpravy existujúceho UI – boli doplnené nové prvky, odstránené identifikované nedostatky a ovládanie niektorých častí bolo sprístupnené intuitívnejším spôsobom. Medzi zapracované zmeny patrili aj nové spôsoby interakcie s prostredím, vyššia úroveň konfigurovateľnosti, ako aj vylepšenia vizualizácie scény a histogramu. Tieto úpravy boli implementované na základe požiadaviek respondentov a vývojových prispievateľov komponentu. Upravená verzia bola následne nasadená do produkčného prostredia systému NDMVR.

Aj keď nebolo možné z organizačných dôvodov realizovať opätovné testovanie s rovnako rozsiahlym a odborným zastúpením respondentov, vykonané zmeny reflektujú konkrétne identifikované nedostatky, čo podporuje ich relevantnosť a zmysluplnosť. Okrem konkrétnych úprav rozhrania sa podarilo pripraviť aj základnú infraštruktúru, ktorá umožní efektívnejšie a v budúcnosti aj automatizované testovanie používateľského zážitku.

Prínosom práce tak nie je len zlepšenie konkrétneho používateľského rozhrania, ale aj podpora celej knižnice jeho súčastí, ktoré môžu byť opätovne použité v rôznych častiach alebo systémoch s podporou A-Frame. Dôležitým výstupom je aj vytvorenie rámca pre ďalší vývoj komponentu NDMVR a UX testovanie, ktorý

zjednodušuje budúcu spätnú väzbu a iteráciu bez potreby zložitej prípravy. Do budúca sa tak otvára priestor na rozšírenie testovania o širšiu skupinu používateľov, zlepšenie analytických nástrojov a prehĺbovanie spolupráce medzi vývojovým tímom a koncovými používateľmi. Zároveň sa tým vytvára stabilné prostredie na udržiavanie konzistentnej používateľskej skúsenosti v rámci celého riešenia.

V práci sa zároveň objavili viaceré otvorené otázky, ktoré presahujú jej rozsah a ktoré by mohli byť predmetom ďalšieho výskumu a vývoja. Medzi ne patrí napríklad hľadanie optimálnych foriem automatizovanej analýzy používateľskej interakcie vo VR, dôslednejšie prispôsobovanie používateľského rozhrania pre špecifické skupiny používateľov (napr. starší účastníci, deti, používatelia so zdravotným znevýhodnením), ako aj vývoj štandardizovaných metrík na hodnotenie používateľskej skúsenosti vo VR prostrediach. Za ďalší odporúčaný krok možno považovať systematické dlhodobé testovanie naprieč rôznymi verziami komponentu.

V kombinácii s pokračujúcim vývojom a úzkou spoluprácou medzi výskumným a vývojovým tímom môžu tieto aktivity významne prispieť nielen k vytvoreniu stabilného a škálovateľného VR riešenia s vysokou používateľskou hodnotou, ale aj k širšiemu porozumeniu problematiky vizualizácie a analýzy dát detektora ALICE. V tejto súvislosti možno konštatovať, že komponent NDMVR má potenciál stať sa dôležitým nástrojom pri výučbe a výskume v oblasti jadrovej fyziky a zároveň prispieť k ďalšiemu rozvoju vzdelávacieho systému smerom k moderným, interaktívnym formám, ktoré poskytujú širší pohľad na skúmanú problematiku.

Literatúra

1. CARSON, Daniel. *Research-Paper-WebGL*. 2020. Dostupné z doi: 10.13140/RG.2.2.18538.95683.
2. PENG, Cui. The research and design Of 3D Web guide system based On WebGL. In: *The 26th Chinese Control and Decision Conference (2014 CCDC)*. 2014, s. 4052–4054. Dostupné z doi: 10.1109/CCDC.2014.6852890.
3. DIRKSEN, Jos. *Learning Three.js – the JavaScript 3D Library for WebGL - Second Edition*. Packt Publishing Ltd, 2015. Dostupné tiež z: http://books.google.ie/books?id=Xja9BwAAQBAJ&pg=PA400&dq=ISBN+978-1-78216-628-3&hl=&cd=1&source=gbs_api.
4. *A-Frame: A web framework for building virtual reality experiences*. [B.r.]. Dostupné tiež z: <https://github.com/aframevr/aframe>. dostupné: 24.3.2024.
5. *WebGL-documentation-site*. [B.r.]. Dostupné tiež z: <https://immersiveweb.dev/>. dostupné: 24.3.2024.
6. KORECKO, Stefan; VALA, M.; FEKETE, M. VISUALIZATION OF EXPERIMENTAL DATA IN WEB- BASED VIRTUAL REALITY. In: 2021, s. 149–152. Dostupné z doi: 10.54546/MLIT.2021.42.97.001.
7. PROKOP, Natanael. *Používateľský zážitok vo virtuálnych prostrediach a jeho vyhodnocovanie*. Košice: Letná 9, 040 01 Staré Mesto, 2023. Technická univerzita v Košiciach, Fakulta elektrotechniky a informatiky. An optional note.
8. CHEN, Songtao; THADURI, Upendar; BALLAMUDI, Venkata. Front-End Development in React: An Overview. *Engineering International*. 2019, roč. 7, s. 117–126. Dostupné z doi: 10.18034/ei.v7i2.662.
9. BRUN, Rene; CANAL, Philippe; RADEMAKERS, Fons. History of the ROOT system: conception, evolution and experience. In: 2011, s. 002. Dostupné z doi: 10.22323/1.093.0002.

10. MASSIMILIANO GALLI, Enric Tejedor; WUNSCH, Stefan. A New PyROOT: Modern, Interoperable and More Pythonic. *EPJ Web of Conferences*. 2020, roč. 245, s. 6. Dostupné z DOI: <https://doi.org/10.1051/epjconf/202024506004>.
11. GONZALEZ MALINE, D.; MONETA, L.; ANTICHEVA, I. New ROOT graphical user interfaces for fitting. *J. Phys. Conf. Ser.* 2010, roč. 219, s. 042010. Dostupné z DOI: [10.1088/1742-6596/219/4/042010](https://doi.org/10.1088/1742-6596/219/4/042010).
12. *ROOT visualization: A dataset fitted with a bidimensional function visualised as a colored surface*. [B.r.]. Dostupné tiež z: <https://root.cern.ch/root/html/doc/guides/primer/figures/fitted2dFunction.png>.
13. ENGEL, Heiko; ALT, Torsten; KEBSCHULL, U. FPGA based data processing in the ALICE High Level Trigger in LHC Run 2. *Journal of Physics: Conference Series*. 2017, roč. 898, s. 032018. Dostupné z DOI: [10.1088/1742-6596/898/3/032018](https://doi.org/10.1088/1742-6596/898/3/032018).
14. MEGY, Théo. *Measurement of the Higgs-boson coupling to tau-leptons and combined measurements of Higgs-boson properties at $\sqrt{s} = 13$ TeV with the ATLAS detector*. 2019. Dostupné z DOI: [10.6094/UNIFR/151139](https://doi.org/10.6094/UNIFR/151139). Diz. pr. Freiburg U., Freiburg U.
15. CHATRCHYAN, S. et al. Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC. *Physics Letters B*. 2012, roč. 716, č. 1, s. 30–61. ISSN 0370-2693. Dostupné z DOI: <https://doi.org/10.1016/j.physletb.2012.08.021>.
16. JSROOT. [B.r.]. Dostupné tiež z: <https://root.cern/manual/jsroot/>.
17. *A-frame.io-documentation-site*. [B.r.]. Dostupné tiež z: <https://aframe.io/docs/1.3.0/introduction/>. dostupné: 24.3.2024.
18. VÄÄNÄNEN-VAINIO-MATTILA, Kaisa; ROTO, Virpi; HASSENZAHN, Marc. Classifying and Selecting UX and Usability Measures. *Meaningful measures: Valid useful user experience measurement (VUUM)*. 2008, s. 13–18.
19. EIDLÖTH, Lisa; MEINERS, Anna-Lena; THOMASCHESKI, Jörg; HINDERKS, Andreas. Pragmatic versus Hedonic: Determining the Dominant Quality in User Experience for Professional and Leisure Collaboration Tools. In: *Proceedings of the 19th International Conference on Web Information Systems and Technologies (WEBIST)*. 2023, s. 391–398. ISBN 978-989-758-672-9. ISSN 2184-3252. Dostupné z DOI: [10.25968/opus-3145](https://doi.org/10.25968/opus-3145).

20. KONEČNÝ, Stanislav. Learning and Education. *Studia Scientifica Facultatis Paedagogicae Universitas Catholica Ružomberok* [online]. 2024, roč. 23, s. 11–22 [cit. 2025-03-08]. Dostupné z doi: 10.54937/ssf.2024.23.1.11-22.
21. GARCÍA, Matías; CANO, Sandra; MOREIRA, Fernando. User eXperience (UX) Evaluation in Virtual Reality (VR). In: ROCHA, Alvaro; ADELI, Hojjat; DZEMYDA, Gintautas; MOREIRA, Fernando; COLLA, Valentina (ed.). *Information Systems and Technologies*. Cham: Springer Nature Switzerland, 2024, s. 207–215. ISBN 978-3-031-45642-8.
22. GIATZOGLOU, Evgenia; VORIAS, Panagiotis; KEMM, Ryan; KARAYIANNI, Irene; NEGA, Chrysanthi; KOURTESIS, Panagiotis. The Trail Making Test in Virtual Reality (TMT-VR): The Effects of Interaction Modes and Gaming Skills on Cognitive Performance of Young Adults. *Applied Sciences*. 2024, roč. 14, č. 21. ISSN 2076-3417. Dostupné z doi: 10.3390/app142110010.
23. ALATHAS, Hadi. How to Measure Product Usability with the System Usability Scale (SUS) Score. 2018. Dostupné tiež z: <https://uxplanet.org/how-to-measure-product-usability-with-the-system-usability-scale-sus-score-69f3875b858f>.
24. LAUBHEIMER, Page. Beyond the NPS: Measuring Perceived Usability with the SUS, NASA-TLX, and the Single Ease Question After Tasks and Usability Tests. 2018. Dostupné tiež z: <https://www.nngroup.com/articles/measuring-perceived-usability/>.
25. O'BRIEN, Heather L.; CAIRNS, Paul; HALL, Mark. A practical approach to measuring user engagement with the refined user engagement scale (UES) and new UES short form. *International Journal of Human-Computer Studies*. 2018, roč. 112, s. 28–39. ISSN 1071-5819. Dostupné z doi: <https://doi.org/10.1016/j.ijhcs.2018.01.004>.
26. DOGAN, Nurhan; DOGAN, Ismet. DETERMINATION OF THE NUMBER OF BINS/CLASSES USED IN HISTOGRAMS AND FREQUENCY TABLES: A SHORT BIBLIOGRAPHY. *TurkStat, Journal of Statistical Research*. 2010, roč. 07, s. 77–86.
27. PEARSON, Karl. Contributions to the Mathematical Theory of Evolution. II. Skew Variation in Homogeneous Material. *Philosophical Transactions of the Royal Society of London. A* [online]. 1895, roč. 186, s. 343–414 [cit. 2023-01-22]. ISSN 02643820. Dostupné z: <http://www.jstor.org/stable/90649>.

28. FEKETE, Martin. Vizualizácia experimentálnych údajov v zdieľanej rozšírenej realite a jej používateľské rozhranie. In: Technická univerzita v Košiciach Fakulta, 2021.
29. MIKULA, Lukáš. Možnosti vizualizácie experimentálnych údajov vo webovej rozšírenej realite. In: Technická univerzita v Košiciach Fakulta, 2022.
30. AFRAMEVR. *Aframe showcase of responsive UI* [online]. GitHub, 2015. [cit. 2024-10-05]. Dostupné z : <https://github.com/aframevr/aframe/blob/master/examples/showcase/ui/index.html>.
31. *Getting Started with the Meta Quest 3: A Beginner's Guide*. VR Wave, 2023. Dostupné tiež z: <https://www.vr-wave.store/blogs/virtual-reality-prescription-lenses/getting-started-with-the-meta-quest-3-a-beginners-guide>.

Zoznam príloh

Príloha A CD médium – záverečná práca v elektronickej podobe,

Príloha B Používateľská príručka

Príloha C Systémová príručka

Obsah prílohy A:

- Adresár src - obsahuje projekt NDMVR so všetkými zdrojovými súbormi
- Adresár doc - obsahuje túto prácu, používateľskú príručku a anonymizované výsledky celého UX testovania vo formáte PDF
- Adresár tex - Obsahuje zdrojové kódy záverečnej práce uvedené v adresári doc

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

**Vylepšenie používateľského rozhrania
virtuálneho prostredia pre vizualizáciu
experimentálnych údajov**

Používateľská príručka

Inštalácia a spustenie projektu NDMVR

1. Spustenie cez verejnú verziu projektu:

- Ak chcete spustiť verejnú verziu projektu, stačí navštíviť poskytnutú URL adresu. Táto stránka zobrazuje všetky aktuálne zmeny vykonané na projekte k dátumu 14. apríla 2025. Môže sa stať, že niektoré z týchto zmien budú v budúcnosti prepísané alebo upravené. Ak chcete používať aktuálnu verziu projektu, odporúčame vykonať lokálne spustenie, ako je uvedené v prílohe A – adresár src.
- Adresa projektu: ¹

2. Spustenie prostredníctvom lokálnej verzie:

- Pred spustením lokálnej verzie projektu NDMVR je potrebné nainštalovať nasledujúce nástroje:
 - NPM (Node Package Manager)
 - Node.js
 - Ngrok
- Postup pre spustenie lokálnej verzie projektu:
 - (a) Skopírujte celý obsah adresára src z prílohy B.
 - (b) Otvorte terminál v hlavnom adresári projektu.
 - (c) Nainštalujte potrebné závislosti projektu pomocou príkazu: `npm install` alebo `sudo npm install` (tento príkaz je potrebný v prípade, že máte problémy s právami administrátora).
 - (d) Spustite vývojovú verziu projektu príkazom: `npm run dev`.
 - (e) Ak je proces úspešný, v terminálovom okne by sa mal zobrazíť port localhostu, na ktorom beží projekt.
- **Spustenie NDMVR pre headset (Oculus Quest 2):**
 - (a) Po spustení localhostu otvorte aplikáciu ngrok.
 - (b) V aplikácii ngrok zadajte príkaz: `ngrok http [číslo portu localhostu]`.
 - (c) Ngrok vygeneruje adresu pre forwardovanie, ktorú zadajte do prehliadača vášho headsetu.

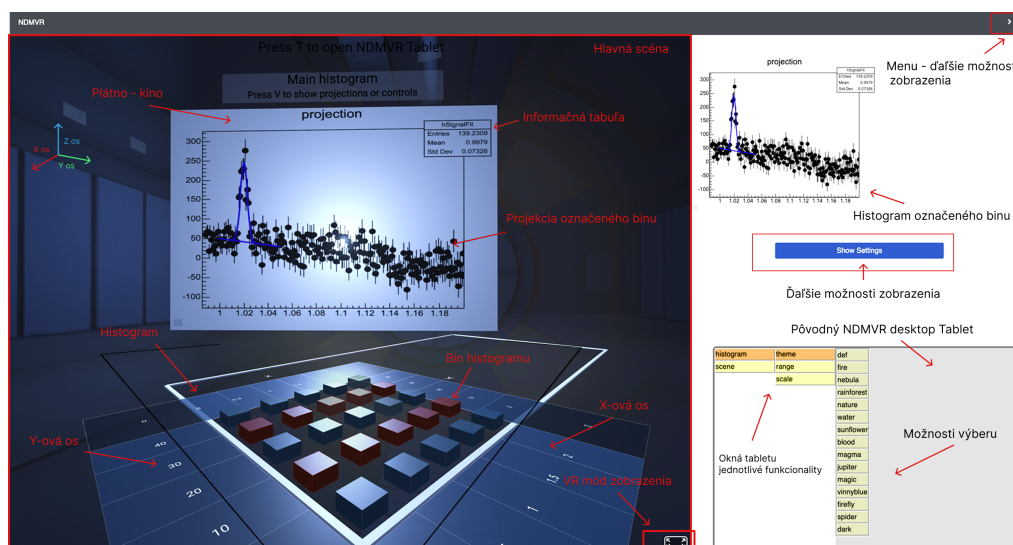
Vďaka nástroju ngrok je možné pristúpiť k projektu NDMVR aj cez zariadenie Oculus Quest 2. Je potrebné, aby bolo vaše zariadenie pripojené k rovnakej

¹<https://ndmspc.gitlab.io/ndmvr>

sieti ako váš počítač. Na spustenie aplikácie v zariadení Oculus Quest 2 použite odkaz vygenerovaný aplikáciou ngrok. Pre ukončenie programu môžete použiť klávesovú skratku Ctrl + C v oboch oknách terminálu alebo aplikácie ngrok.

Základného zobrazenie NDMVR a jeho možnosti

Na obrázku nižšie je zobrazené základné zobrazenie spolu s vysvetlením.



Základné zobrazenie NDMVR spolu s vysvetlením. (zdroj: NDMVR)

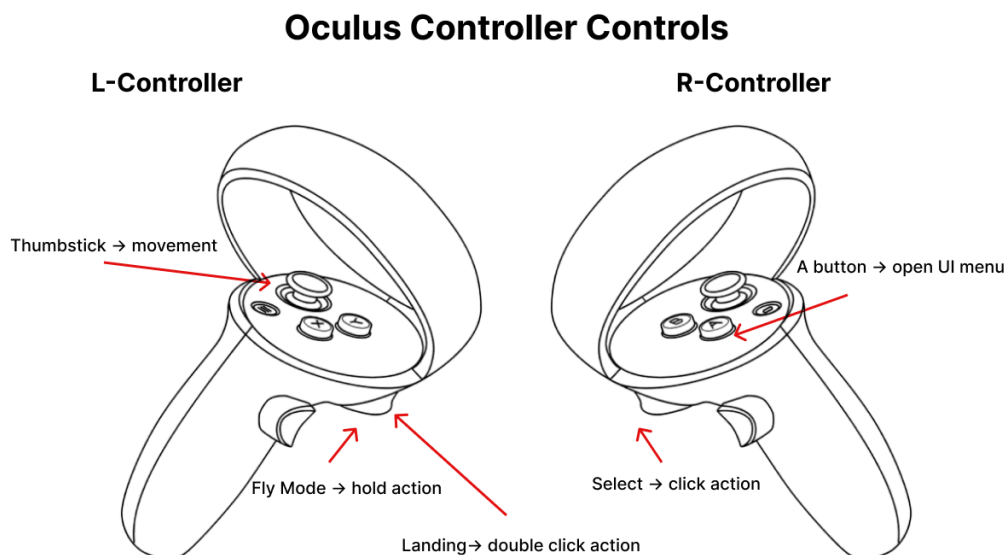
Menu NDMVR ponúka nasledujúce zobrazenia:

- DYNAMIC TH3 histogram example
- GITLAB data projection example
- PHYSICS data Analysis projection example (zobrazenie použité v rámci scenára testovania – upravená verzia špeciálne pre tieto účely)
- PHYSICS data projection example
- NDMVR monitoring adaptation
- NDMVR Shared Projection example

Zobrazenia *PHYSICS data projection example*, *GITLAB data projection example* a *PHYSICS data Analysis projection example* sú vybavené aktualizáciami histogramu a binov. Verzia TH3 momentálne nie je podporovaná, preto pri používaní nástroja odporúčame vybrať jedno z týchto zobrazení pre optimálny zážitok.

Ovládanie implementovaných častí NDMVR

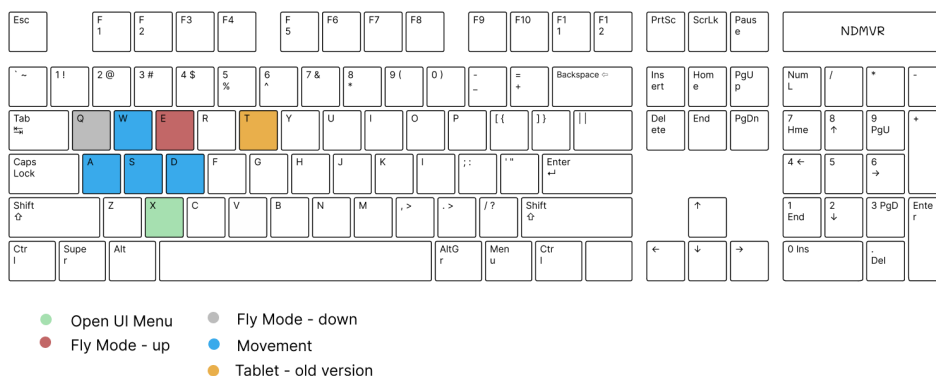
Obrázok zobrazuje základné prvky ovládania pre VR zobrazenie v zariadení Oculus.



Prvky ovládania pre VR zobrazenie v zariadení Oculus. (zdroj: NDMVR)

**Pre správne zobrazenie verzie NDMVR pre Oculus Quest 2 je nevyhnutné vstúpiť do režimu VR.*

Desktop Controls



Ovládanie pre desktopvého zobrazenie. (zdroj: NDMVR)

Nasledujúci obrázok zobrazuje základné prvky ovládania pre VR režim desk-

topvého zobrazenie.

**Pre zobrazenie verzie NDMVR na celú obrazovku je nevyhnutné vstúpiť do režimu VR.*

Spustenie vývojárskeho režimu

Najjednoduchším spôsobom, ako spustiť vývojárske zobrazenie, je použitie aplikácie Meta Quest Developer Hub (MQDH). Táto aplikácia je dostupná pre operačné systémy Windows ² aj macOS ³.

Aby bolo možné otvoriť konzolu, je potrebné vykonať nasledujúce kroky ⁴:

1. Konfigurácia MQDH a zariadenia Oculus:

1. Spustíte aplikáciu MQDH.
2. Nastavíte správnu cestu k ADB (Android Debug Bridge).
3. Prihlásite sa pomocou vývojárskeho účtu.
4. Ak účet ešte nie je overený, vykonajte overenie (je to nevyhnutné pre aktiváciu vývojárskeho režimu).
5. Pripojíte zariadenie Oculus k počítaču pomocou USB-C kábla.
6. Po zobrazení výzvy na zariadení Oculus povoľte prístup k údajom.
7. V aplikácii MQDH pridajte nové zariadenie.
8. Zo zoznamu zariadení vyberte svoje zariadenie (napr. Ndm Spc).
9. Skontrolujte, či je vývojársky režim aktívny.

2. Možnosti pripojenia:

1. Pripojenie cez USB (odporúčané, rýchlejšie)
2. Pripojenie cez WiFi (pomalšie)

(a). Pripojenie cez ADB a USB:

1. V prehliadači Google Chrome otvorte stránku: *chrome://inspect/#devices*.
2. Aktivujte možnosť „Objaviť zariadenia USB“.
3. Vyberte predvolenú možnosť.
4. Zo zoznamu kariet otvorených na zariadení Oculus vyberte kartu s názvom NDMVR.

²<https://developer.oculus.com/downloads/package/oculus-developer-hub-win/>

³<https://developer.oculus.com/downloads/package/oculus-developer-hub-mac/>

⁴<https://developer.oculus.com/documentation/web/browser-remote-debugging/>

5. Kliknite na tlačidlo „Inspect“ pre otvorenie vývojárskej konzoly pre NDMVR.

(b). Pripojenie cez ADB a WiFi:

1. V aplikácii MQDH zapnite ADB cez WiFi.
2. V MQDH alebo príkazovom riadku (cmd) spustíte príkaz: `adb shell ip route`.
3. V prehliadači Google Chrome otvorte stránku: `chrome://inspect/#devices`.
4. Aktivujte možnosť „Objaviť ciele siete“.
5. Zadajte IP adresu vygenerovanú ADB príkazom a predvolený port: 5555.
6. Zo zoznamu otvorených kariet na zariadení Oculus vyberte kartu NDMVR.
7. Kliknite na tlačidlo „Inspect“ pre otvorenie vývojárskej konzoly pre NDMVR.

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

**Vylepšenie používateľského rozhrania
virtuálneho prostredia pre vizualizáciu
experimentálnych údajov**

Systémová príručka

UI Menu - možnosti konfigurácie

Táto dokumentácia popisuje štruktúru JSON konfigurácie pre celý UI menu, od navigácie v paneli kariet až po konkrétne pohľady v okne menu (obsah, alebo preddefinované šablóny s rozložením a štýlom, ktoré môžu byť voľne modifikované na základe konfigurácie). Navigácia v paneli kariet obsahuje rôzne komponenty, ktoré sa aktivujú po kliknutí na konkrétnu ikonu. Komponenty zahŕňajú inštrukcie, hodnotiace otázky, hudobný prehrávač a nastavenia. Štruktúra navigácie v paneli kariet, ako aj každý komponent (šablóna), má svoje vlastnosti, ktoré sú podrobne vysvetlené nižšie spolu s príkladmi použitia.

TabBarNavigation konfigurácia

Tento komponent definuje obsah navigačného panela s rôznymi položkami menu. Každá položka menu má špecifické nastavenia, ako farby, obrázky, akcie a komponenty.

Príklad štruktúry

```
{
  "tabBarNavigationComponentContent": {
    "scheme": "navigation_tab_bar",
    "description": "Navigation bar selector:",
    "properties": {
      "menuItems": {
        "type": "select",
        "options": [
          {
            "color": "#bf8f8c",
            "src": "menu/home_24dp.png",
            "width": "1",
            "height": "1",
            "radius": "0.2",
            "activeColor": "#FFEB3B",
            "backgroundColor": "#F9FBE7",
            "hoverColor": "#F0F4C3",
            "borderColor": "#827717",
            "fontColor": "#000",
```

```

        "onclick": "loadSusQuestions()",
        "component": "InstructionArea"
    }
    ...

```

Vysvetlenie polí

- **menuItems** : Zoznam položiek menu, každá s nasledujúcimi vlastnosťami:
 - **color** : Definuje farbu položky menu v hex formáte.
 - **src** : Cesta k obrázku ikony.
 - **width, height** : Rozmery položky menu.
 - **radius** : Polomer zaoblenia okrajov položky.
 - **activeColor, backgroundColor, hoverColor** : Farby pre rôzne stavy položky.
 - **borderColor** : Farba okraja položky.
 - **fontColor** : Farba textu pre popisy alebo popisky.
 - **onclick** : JavaScript funkcia spustená pri kliknutí na položku.
 - **component** : Pridružený komponent, ktorý sa zobrazuje po výbere.

Použitie

Ak kliknete na položku menu s danou icon, spustí sa funkcia definovaná v `onclick` a otvorí sa komponent `InstructionArea`.

To Do Template konfigurácia

Definuje krok za krokom inštrukcie pre rôzne scenáre. (Šablóna číslo 1)

Príklad štruktúry

```

{
  "instructionsComponentContent": {
    "scheme": "to_do_list",
    "description": "Instructions area:",
    "properties": {
      "instructions": {

```

```

    "type": "select",
    "options": [
      {
        "id": "container-window-1",
        "title": "Analyza c.1",
        "lines": [
          "1. task",
          "2. task",
          "3. task"
        ]
      }
    ]
  }
  ...

```

Vysvetlenie polí

- **instructions** : Objekt inštrukcií, každá inštrukcia obsahuje:
 - **id** : Jedinečný identifikátor pre súbor inštrukcií.
 - **title** : Názov popisujúci súbor inštrukcií.
 - **texts** : Krok za krokom textové inštrukcie.

Použitie

Inštrukcia 'Analysis no.1' obsahuje tri kroky, ktoré používateľovi ukazujú, ako zväčšiť histogram a analyzovať obsah binu.

Questionnaire Template konfigurácia

Komponent je používaný na zobrazenie otázok a možných odpovedí, ich zaznamenanie a odoslanie údajov do úložiska. (použitie šablóny číslo 2)

Príklad štruktúry

```

{
  "susComponentContent": {
    "scheme": "feedback_form",
    "description": "UES questionnaire:",
    "properties": {
      "form": {

```

```

    "type": "select",
    "options": [
      {
        "question": "I think that ...",
        "answers": {
          "type": "select",
          "options": [
            {
              "value": "Strongly \nDisagree",
              "image": "menu/looks_1_24dp.png"
            }
          ]
        }
      }
    ]
  }
  ...

```

Vysvetlenie polí

- **questions** : Pole objektov, kde každý objekt predstavuje otázku. Každá otázka obsahuje nasledujúce:
 - **text** : Text otázky zobrazený používateľovi.
- **options** : Pole objektov, ktoré reprezentujú možné odpovede na otázky v poli questions.
 - **value** : Textová značka pre možnosť odpovede.
 - **image** : Cesta k obrázku, ktorý vizuálne zobrazuje možnosť odpovede.

Použitie

Používateľ je požiadaný o poskytnutie spätnej väzby.

Music Player Template konfigurácia

Šablóna číslo 3 - Tento komponent umožňuje prehrávanie hudby. Obsahuje zoznam skladieb, z ktorých si používateľ môže vybrať a prehrať danú položku. Je tu preddefinované rozloženie pre zobrazenie obsahu v 3x3 mriežke.

Príklad štruktúry

```
{
  "musicPlayerComponentContent": {
    "scheme": "music_player",
    "description": "",
    "properties": {
      "songs": {
        "type": "select",
        "options": [
          {
            "title": "Relaxing Forest \nMorning",
            "description": "Music designed for ...",
            "url": "/menu-songs/music.mp3",
            "color": "#8783D1"
          }
        ]
      }
    }
  }
  ...
}
```

Vysvetlenie polí

- **songs** : Pole skladieb, kde každá skladba je objekt s nasledujúcimi vlastnosťami:
 - **title** : Názov skladby, ktorý môže obsahovať zalomenia riadkov.
 - **description** : Krátky popis skladby, ktorý môže obsahovať zalomenia riadkov.
 - **url** : Cesta k audio súboru skladby.
 - **color** : Farebný kód spojený so skladbou, používaný na zobrazenie v UI.

Použitie

Túto štruktúru je možné použiť v rôznych kontextoch, ako prehrávač hudby, kde si používatelia môžu vybrať skladby na prehrávanie, alebo ako pozadie pre špecifické scény či aktivity.

Settings Template konfigurácia

Tento komponent obsahuje nastavenia aplikácie, kde si používateľ môže vybrať rôzne možnosti ako tému, scénu, rozsah alebo mierku.

Príklad štruktúry

```
{
  "settingsContent": {
    "scheme": "settings",
    "description": "Options:",
    "optionsPerPage": 15,
    "propertiesPerPage": 4,
    "selectFirstPropertyOnPageChange": true,
    "properties": {
      "theme": {
        "scheme": "histogram",
        "description": "Choose a theme: ",
        "type": "select",
        "value": "def",
        "options": [
          "def"
        ]
      },
      "scene": {
        "scheme": "scene",
        "description": "Choose a scene: ",
        "name": "scene",
        "value": "ndmvr",
        "type": "select",
        "options": [
          "ndmvr"
        ]
      },
      ...
    }
  }
}
```

Vysvetlenie polí

- **settingsContent** : Objekt obsahujúci obsah nastavení.
 - **description** : Všeobecný popis nastavení na vrchole (text, ktorý je zobrazený).

- **properties** : Objekt obsahujúci konkrétne nastavenia, ako napríklad tému, scenár alebo mierku.
- **type** : Typ ovládacieho prvku, napríklad `select` pre výber.
- **options** : Možnosti na výber pre dané nastavenie.

Použitie

Tento komponent poskytuje používateľovi rozhranie na zmenu nastavení aplikácie.

Pridanie vlastnej akcie šablóny

Pre každú šablónu a jej vlastnosti (`property`) je možné pridať vlastnú akciu, ktorá bude spustená pri interakcii s komponentom. Vlastná akcia je definovaná ako reťazec v atribúte `'action'`, ktorý sa pridáva do príslušnej šablóny v JSON konfigurácii.

Príklad priradenia vlastnej akcie

```
"action": "customActionName"
```

Akcia môže byť akýkoľvek názov funkcie alebo metódy, ktorá bude spracovávať túto udalosť. Tento názov akcie bude následne použitý na spustenie špecifickej logiky alebo správania.

Prihlásenie sa na odber akcie

Ak chcete sledovať udalosť, ktorá je spojená s konkrétnou akciou, môžete využiť tento postup:

Postup sledovania akcie

```
// Prihlásenie sa na odber udalosti
const subscription = emitter
  .addListener('cstomActionName', () => {
    console.log('My action running...');
    // add implementation
  });

// Odhlásenie sa z odberu udalosti
subscription.remove();
```

Tvorba a správa UI-menu v Reacte

Komponent: Menu

Popis

Komponent Menu je hlavným rendrovacím prvkom. Poskytuje konfiguráciu pre daný React element (šablónu) spolu s jeho komponentami a obsahom pre dynamické renderovanie, a následne ho odosiela do časti tabBar navigácie, ktorá zodpovedá za zobrazenie jednotlivých obsahov okien UI menu. Tento komponent sa prispôsobuje rôznym zobrazeniam a je najvyšším komponentom v stromovej, teda hierarchickej štruktúre UI menu. Zaobaluje Reactovskú časť kombinovanú s prvkami poskytnutými A-frame knižnicou pre UI menu. Komponent je kombinovaný s raycasterom [gui-interactable], nakoľko tento raycaster, spolu s .collidable, sú súčasťou primitív knižnice A-Frame a poskytujú možnosť aktivovania interakcií a ich zachytenia.

Použitie (React JSX)

```
<a-entity
  id="cameraRig"
  cursor="rayOrigin:
mouse; fuse: false"
  raycaster="objects: .clickable,
    [gui-interactable], .collidable;far:5500">
  // Main camera
  <a-camera
    ...
  </a-camera>

  // Menu entity
  <a-entity fixed-position>
```

```

    <Menu/>
  </a-entity>
</a-entity>

```

Popis Parametrov

Názov	Typ	Popis
configurationFile	object	Konfiguračný súbor obsahujúci nastavenia pre rôzne komponenty, ako sú napríklad pokyny, formuláre spätnej väzby, nastavenia a hudobný prehrávač.
selectedComponent	string	Kľúč aktuálne zvoleného komponentu, ktorý sa má zobrazíť v hlavnom okne. Predvolená hodnota je 'ToDoList'.
tabBarConfiguration	object	Konfigurácia pre navigačný panel, ktorá určuje obsah zobrazený v tabbare. Obsahuje informácie o rôznych navigačných komponentoch.

Použité časti A-Frame

- **a-entity** – Hlavný kontajner pre celé VR menu. Tento element zobrazuje všetky komponenty, ako aj navigáciu a klávesnicu.
- **a-xywindow** – Komponent na zobrazenie okna s komponentami, ktoré sú vizualizované v 3D priestore.
- **a-xykeyboard** – Virtuálna klávesnica na zadávanie textu.

Použité časti React

- **TabBarNavigation** – Komponent na zobrazenie navigačného panelu s možnosťou prepínania medzi rôznymi sekciami aplikácie.

Interné mechanizmy a funkcie

- **useNdmVrRedirect** – hook na získanie informácií o VR zariadení, ako je Oculus, ktorý sa používa na prispôsobenie pozície a veľkosti okna.

- **useState** – hook na uchovávanie stavu aktuálne zvoleného komponentu (`selectedComponent`).
- **useEffect** – hook, ktorý sa používa na subscribovanie na `schemaSubject` na získanie konfiguračných údajov pre komponent `Settings`.
- **renderComponent** – funkcia na vykreslenie vybraného komponentu na základe aktuálneho stavu `selectedComponent`. Získa sa aj konfigurácia pre daný komponent zo stavu `componentProps`.

Komponent: TabBarNavigation

Popis

Komponent `TabBarNavigation` je zodpovedný za zobrazenie navigačného panela, ktorý umožňuje prepínanie medzi rôznymi komponentmi menu. Používa obsah dynamicky generovaný z konfiguračného súboru, čo umožňuje používateľom upravovať vzhľad a akcie komponenty umiestnenej na navigačnom paneli.

Použitie (React JSX)

```
<TabBarNavigation  
  onSelect={callbackFunction}  
  configurationFile={menuConfigFile}  
>
```

Popis Parametrov

Názov	Typ	Popis
<code>onSelect</code>	<code>function</code>	Funkcia, ktorá sa zavolá pri výbere položky na navigačnom paneli. Tento callback funkcia sa použije na zobrazenie vybraného komponentu.
<code>configurationFile</code>	<code>object</code>	Konfiguračný súbor, ktorý obsahuje možnosti pre položky menu, ako sú názvy, farby a ikony. Tento súbor definuje, aké komponenty sa zobrazia v navigačnom paneli.

Použité časti A-Frame

- **a-entity** – hlavný kontajner, ktorý drží celé menu a umožňuje navigáciu medzi jednotlivými komponentmi.
- **a-xycontainer** – obalový kontajner, ktorý zobrazuje jednotlivé položky v menu. Tieto položky sú reprezentované ako ikony s rôznymi vizuálnymi vlastnosťami.
- **a-menu-icon** – ikona pre každú položku menu. Táto ikona môže byť prispôbená podľa konfiguračného súboru, s možnosťami ako sú farby, obrázky a rôzne vizuálne efekty.

Interné mechanizmy a funkcie

- **useNdmVrRedirect** – hook, ktorý sa používa na prispôbenie vzhľadu a pozície menu na základe typu VR zariadenia (napríklad Oculus).
- **useState** – hook používaný na uchovávanie stavu položiek menu (`menuItems`), ktoré sú získané z konfiguračného súboru.
- **useEffect** – hook, ktorý sa používa na načítanie konfiguračného súboru a nastavenie položiek menu pri načítaní komponentu.
- **window.openComponent** – funkcia, ktorá je dostupná globálne a zavolá funkciu `onSelect` pri kliknutí na položku menu, čo spôsobí zobrazenie požadovaného komponentu.
- **renderovanie položiek menu** – dynamické generovanie ikón podľa konfiguračného súboru. Každá ikona je prispôbená podľa hodnoty, ako sú farby, obrázky a ďalšie vizuálne vlastnosti.
- **onSelect callback** – funkcia, ktorá sa zavolá, keď používateľ klikne na položku menu. Tento callback sa používa na zmenu aktuálne zobrazeného komponentu v aplikácii.

Komponent: SelectOptions

Popis

Komponent `SelectOptions` poskytuje interaktívne rozhranie na výber a konfiguráciu rôznych možností zo súboru konfigurácie. Komponent umožňuje používateľovi vybrať rôzne kategórie, ktoré sú potom zobrazené v pravom paneli spolu s možnosťami (napr. číselné alebo výberové možnosti). Komponent tiež obsahuje vlastné navigačné tlačidlá pre stránkovanie a výber medzi rôznymi položkami.

Použitie (React JSX)

```
<SelectOptions
  configurationFile={configurationFile}
  schemaConfiguration={schemaConfiguration}
/>
```

Tento príklad zobrazuje komponent `SelectOptions`, ktorý je pripojený ku konfigurácii a schéme, ktoré sú odovzdané ako parametre. Komponent vykreslí interaktívne rozhranie s možnosťami na výber a navigáciu.

Popis Parametrov

Názov	Typ	Popis
<code>configurationFile</code>	object	Objekt, ktorý obsahuje konfiguračné možnosti a popisy, vrátane rôznych kategórií a schém pre daný výber.
<code>schemaConfiguration</code>	object	Objekt, ktorý obsahuje schému konfigurácie, ktorá bude použitá na zobrazenie a aktualizáciu možností.

Funkcie

Pohyb medzi stránkami

- **handlePrevious** – funkcia, ktorá aktualizuje stav aktuálnej stránky pri navigácii na predchádzajúcu stránku.
- **handleNext** – funkcia, ktorá aktualizuje stav aktuálnej stránky pri navigácii na ďalšiu stránku.

- **currentPage** – uchováva informáciu o aktuálnej stránke, ktorá sa zobrazuje v pravom paneli.
- **itemsPerPage** – určuje, koľko položiek sa zobrazuje na jednej stránke.

Generovanie unikátnych kľúčov

- **generateUniqueKey** – generuje jedinečný kľúč, ktorý je priradený každému z komponentov v rámci stránky.

Výber kategórie a schémy

- **handleSchemeChange** – funkcia, ktorá aktualizuje hodnoty v schéme, keď používateľ zvolí novú hodnotu pre vybranú kategóriu.
- **selectedCategory** – uchováva informáciu o aktuálne vybranej kategórii.
- **selectedScheme** – uchováva informáciu o aktuálne vybranej schéme pre danú kategóriu.
- **handleSchemeChange** – aktualizuje schému pri zmene hodnoty vybranej kategórie alebo schémy.
- **executeCustomCallback(value)** – Ak je v konfigurácii definovaný vlastný callback, odošle udalosť prostredníctvom `EventEmitter` s hodnotou (vybranej kategórie a získanej hodnoty) ako parametrom.

Renderovanie ľavého panelu

- **renderLeftPanel** – Funkcia, ktorá zobrazuje kategórie (napr. témy alebo scény) v ľavom paneli. Každá kategória je zobrazená podľa svojho typu – `Select` alebo `Range`.

Renderovanie pravého panelu

- **renderSelectTypePanel** – funkcia, ktorá vykresľuje panel pre výberové možnosti (`select`), ktorý je zobrazený na pravej strane.
- **renderNumberTypePanel** – funkcia, ktorá vykresľuje panel pre číselné možnosti (`range`), ktoré sú špecifické pre danú kategóriu.

Použité časti A-Frame

Komponent `SelectOptions` je použitý v A-Frame prostredí na interaktívne zobrazenie a navigáciu medzi rôznymi konfiguračnými možnosťami. Zobrazuje dva panely: ľavý panel pre kategórie a pravý panel pre možnosti.

- `a-xycontainer` – obalová entita pre jednotlivé sekcie v rámci komponentu.
- `a-entity` – obal pre interaktívne tlačidlá a iné obsahové prvky.
- `a-xyimage` – používa sa na zobrazenie ikon alebo tlačidiel.
- `a-text` – textové prvky, ktoré zobrazujú popisy a názvy kategórií alebo možností.

Funkčnosť Navigácie

Komponent využíva vlastné navigačné tlačidlá na stránkovanie cez položky, ktoré sú zobrazené v pravom paneli. Tieto tlačidlá sú implementované pomocou komponentov `LeftPaginator` a `RightPaginator`, ktoré umožňujú používateľovi presúvať sa medzi stránkami.

Komponent: `NumberTypePanel`

Popis

Komponent `NumberTypePanel` umožňuje používateľovi interaktívne upravovať hodnotu v rámci určitého rozsahu pomocou dvoch tlačidiel: jedno na zväčšenie hodnoty a druhé na zmenšenie. Komponent poskytuje dynamickú spätnú väzbu na základe interakcií používateľa, pričom každá zmena hodnoty vyvolá volanie spätného volania `onScaleChange`, ktoré môže aktualizovať iné komponenty alebo štruktúry údajov.

Použitie (React JSX)

```
<NumberTypePanel
  initScale={1.0}
  initStepScale={0.1}
  detailStepScale={0.05}
  detailStartScale={1.0}
  minScale={0.5}
```

```

maxScale={5.0}
onScaleChange={handleScaleChange}
/>

```

Tento príklad zobrazuje komponent `NumberTypePanel` so zadanými parametrami, ktoré určujú počiatočnú hodnotu, kroky zmeny a rozsah. Komponent vykresľuje ovládacie prvky na úpravu mierky v 3D priestore a aktualizuje hodnotu pri každej interakcii.

Popis Parametrov

Názov	Typ	Popis
<code>initScale</code>	number	Počiatočná hodnota mierky, ktorá sa zobrazuje na začiatku.
<code>initStepScale</code>	number	Krok, o ktorý sa hodnota zväčší alebo zmenší pri stlačení tlačidiel.
<code>detailStepScale</code>	number	Podrobnejší krok zmeny, ktorý sa použije, keď je hodnota menšia než počiatočná hodnota.
<code>detailStartScale</code>	number	Počiatočná hodnota, pod ktorou sa použije podrobnejší krok.
<code>minScale</code>	number	Minimálna hodnota mierky, ktorú možno nastaviť.
<code>maxScale</code>	number	Maximálna hodnota mierky, ktorú možno nastaviť.
<code>onScaleChange</code>	function	Spätné volanie, ktoré sa vykoná, keď používateľ zmení mierku.

Funkcie

Manipulácia s hodnotou

Komponent používa dve hlavné operácie: zväčšenie a zmenšenie hodnoty pomocou tlačidiel:

- **Zväčšenie hodnoty** – Keď používateľ stlačí tlačidlo pre zväčšenie *imagePlus*, hodnota sa zväčší podľa preddefinovaného kroku, pokiaľ nie je prekročená maximálna hodnota.

- **Zmenšenie hodnoty** – Keď používateľ stlačí tlačidlo pre zmenšenie *image-Minus*, hodnota sa zmenší podľa preddefinovaného kroku, pokiaľ nie je dosiahnutá minimálna hodnota.

Aktualizácia hodnoty

Hodnota je automaticky aktualizovaná a následne je volané spätné volanie **on-ScaleChange**, ktoré umožňuje zdieľanie tejto hodnoty s inými komponentami.

Nastavenie hodnoty

- **scaleValue** – Uchováva aktuálnu hodnotu mierky, ktorá je zobrazená v strede panelu.
- **setCounter** – Funkcia, ktorá aktualizuje hodnotu mierky na základe akcie ('plus' alebo 'minus') a zabezpečuje, že sa hodnota pohybuje v rámci definovaného rozsahu.
- **Dynamické zobrazenie** – Aktuálna hodnota sa dynamicky zobrazuje na strednom paneli medzi tlačidlami.

Použité časti A-Frame

- **a-xycontainer** – Kontajner pre celkovú štruktúru komponentu, ktorý obsahuje tlačidlá a text.
- **a-xyimage** – Používa sa na vykreslenie obrázkov (tlačidlá na zväčšenie a zmenšenie) v 3D priestore.
- **a-text** – Textový prvok na zobrazenie aktuálnej hodnoty mierky.
- **a-entity** – Obalové entity na umiestnenie a vizualizáciu rôznych prvkov ako pozadia, obrázky a text.

Komponent: SelectTypePanel

Popis

Komponent `SelectTypePanel` umožňuje používateľom vyberať z možností, ktoré sú rozdelené do troch stĺpcov s možnosťou konfigurovať počet položiek na stĺpec. Tieto možnosti sú načítané na základe vybranej kategórie. Komponent zároveň implementuje stránkovanie, ktoré zobrazuje len časť možností na jednej stránke.

Použitie (React JSX)

```
<SelectTypePanel
  selectedCategory={selectedCategory}
  configurationFile={configurationFile}
  currentPage={currentPage}
  itemsPerPage={itemsPerPage}
  onItemChange={handleItemChange}
/>
```

Tento príklad zobrazuje komponent `SelectTypePanel`, ktorý načíta možnosti z kategórie `theme` a zobrazuje ich na prvej stránke. Každá položka je reprezentovaná tlačidlom, ktoré spustí akciu po kliknutí.

Popis Parametrov

Názov	Typ	Popis
<code>selectedCategory</code>	<code>string</code>	Kategória, ktorá bola vybraná pre zobrazenie možností (napr. <code>theme</code> alebo <code>scene</code>).
<code>configurationFile</code>	<code>object</code>	Konfiguračný objekt, ktorý obsahuje vlastnosti a možnosti pre každú kategóriu.
<code>currentPage</code>	<code>number</code>	Aktuálna stránka, ktorá sa má zobraziť.
<code>itemsPerPage</code>	<code>number</code>	Počet položiek na jednu stránku.
<code>onItemChange</code>	<code>function</code>	Spätné volanie, ktoré sa vykoná, keď používateľ vyberie položku.

Funkcie

- **Stránkovanie** – Možnosti sú stránkované a zobrazené po stranách.
- **Výber položky** – Kliknutím na tlačidlo sa vyberie položka a spustí spätné volanie `onItemChange`.
- **Dynamické stĺpce** – Možnosti sú rozdelené do troch stĺpcov, čo zlepšuje prehľadnosť a interaktivitu.

Manipulácia s výberom

Komponent umožňuje používateľovi vybrať jednu z možností, ktorá sa následne spracuje pomocou funkcie `onItemChange`.

- **handleSelect** – Funkcia, ktorá sa spustí pri kliknutí na položku v zozname. Táto funkcia volá spätné volanie `onItemChange` s hodnotou vybranej položky.

Stránkovanie

Položky sú stránkované a zobrazujú sa po stranách na základe parametra **currentPage**. Na stránke sa zobrazuje maximálne **itemsPerPage** položiek. Zobrazenie položiek sa vykonáva v troch stĺpcoch.

Rozdelenie položiek do stĺpcov

Položky sa dynamicky rozdelia do troch stĺpcov. Každý stĺpec obsahuje maximálne 5 položiek, ktoré sú zobrazené ako tlačidlá. Každé tlačidlo umožňuje používateľovi kliknúť a vybrať danú možnosť.

Použité časti A-Frame

- **a-xycontainer** – Kontajner, ktorý drží všetky položky v stĺpci.
- **a-entity** – Obalové entity na vykreslenie tlačidiel a ostatných prvkov.
- **a-gui-button-rounded** – Tlačidlo, ktoré zobrazuje názov položky a spúšťa funkciu `handleSelect` pri kliknutí.

Komponent: FeedbackForm

Popis

Komponent `FeedbackForm` je určený na zobrazenie interaktívneho formulára spätnej väzby v prostredí A-Frame. Tento komponent umožňuje zobrazovať otázky so špecifickými odpoveďami a umožňuje používateľovi vybrať odpoveď, ktoré sa následne pošlú na server prostredníctvom emailu. Komponent využíva rôzne prispôsobiteľné A-Frame komponenty na zobrazenie otázok, odpovedí a interaktívneho ovládania.

Použitie (React JSX)

```
<FeedbackForm
```

```
  configurationFile={configurationFile}
```

```
  componentKey="unique-key"
```

/>

Parametre

Názov	Typ	Popis
configurationFile	object	Konfiguračný súbor obsahujúci možnosti otázok a odpovedí pre formulár. Tento súbor by mal obsahovať vlastnosť <code>form</code> s možnosťami <code>options</code> a <code>answers</code> .
componentKey	string	Unikátny kľúč komponentu, ktorý sa používa na uloženie údajov do lokálneho úložiska (<code>localStorage</code>) pre sledovanie počtu zobrazených formulárov.

Použité časti A-Frame

- **a-xycontainer** – kontajner pre zobrazenie otázok a odpovedí v 3D priestore. Používa sa na zoskupenie komponentov do vertikálne alebo horizontálne orientovaných sekcií.
- **a-text** – zobrazuje textové informácie ako otázky a odpovede.
- **a-xyimage** – zobrazuje obrázky v odpovediach, ktoré sú interaktívne a umožňujú výber odpovede.
- **a-entity** – univerzálny komponent na pridávanie rôznych objektov do A-Frame scény, vrátane komponentov ako `xyitem` a `xycontainer`.
- **a-xywidget** – komponent na zobrazovanie interaktívnych widgetov v scéne.
- **a-xyinput** – komponent na zadávanie textových údajov, ak je potrebné, tento komponent môže byť pridaný pre pokročilý vstup.
- **a-xyimage** – komponent na interaktívny výber odpovedí zobrazený ako obrázok.

Použité časti React

- **LeftPaginator, RightPaginator** – komponenty na navigáciu medzi otázkami formulára, umožňujúce používateľovi prechádzať medzi jednotlivými otázkami.

Interné mechanizmy a funkcie

- **useEffect** – hook na inicializáciu otázok a odpovedí po načítaní konfiguračného súboru.
- **handleNext** – funkcia na prechod na nasledujúcu dvojicu otázok.
- **handlePrevious** – funkcia na návrat na predchádzajúcu dvojicu otázok.
- **handleOptionClick** – funkcia na spracovanie kliknutia na odpoveď, ktorá uloží výber do stavu.
- **handleSendEmail** – funkcia na prípravu a odoslanie emailu so zodpovedanými otázkami.
- **preparedEmail** – funkcia na prípravu parametrov emailu pred jeho odoslaním, keď sú všetky otázky zodpovedané.
- **handleEmailSuccess** – funkcia na spracovanie úspešného odoslania emailu, vrátane aktualizácie počítadla pre zaslané formuláre.
- **renderQuestion** – funkcia na renderovanie jednotlivých otázok v 3D prostredí s priradenými odpoveďami.
- **renderOptions** – funkcia na renderovanie možných odpovedí na otázku.
- **executeCustomCallback(value)** – Ak je v konfigurácii definovaný vlastný callback, odošle udalosť prostredníctvom `EventEmitter` s hodnotou (vykonanej akcie a získanej hodnoty, napríklad odoslanie odpovedí) ako parametrom.

Poznámky

- Vstupy sú spracované a uchovávané v stave `responses` a `questions`.
- Po zodpovedaní všetkých otázok, je možnosť poslať vyplnený formulár emailom v tvare CSV.

Komponent: `ToDoList`

Popis

Komponent `ToDoList` je určený na zobrazenie zoznamu úloh a poskytuje interakciu pre používateľov na zaznamenávanie času vykonávania úloh. Umožňuje

používateľom spustiť časovač, kliknúť na položky zoznamu a odoslať výsledky na email. Tento komponent tiež používa animácie pre vizuálne prvky, ako je časovač.

Použitie (React JSX)

```
<ToDoList configurationFile={todoConfigFile} />
```

Popis Parametrov

Názov	Typ	Popis
configurationFile	object	Konfiguračný súbor, ktorý obsahuje dáta pre jednotlivé úlohy v zozname. Tento súbor určuje názvy úloh, texty a ďalšie nastavenia pre každú položku zoznamu.

Použité časti A-Frame

- **a-entity** – hlavný kontajner pre celý zoznam úloh, obsahujúci vizuálne elementy ako tlačidlá a textové položky.
- **a-xycontainer** – kontajner pre jednotlivé položky úloh a vizuálne prvky, ktorý zabezpečuje správne umiestnenie a organizáciu.
- **a-text** – používané na zobrazenie názvov a popisov úloh v zozname.
- **a-xyimage** – používané na zobrazenie interaktívnych tlačidiel, ktoré spúšťajú akcie ako spustenie časovača alebo odoslanie emailu.

Interné mechanizmy a funkcie

- **useState** – hook používaný na uchovávanie stavu, ako je počet dokončených úloh, časy, ktoré boli merané pre jednotlivé úlohy, a stav časovača.
- **useEffect** – hook, ktorý načítava konfiguračný súbor a načítava údaje zo LocalStorage pre ukladanie stavu používateľa.
- **calculateHeight** – pomocná funkcia na výpočet výšky pre každú položku zoznamu podľa počtu riadkov textu.

- **window.clickAction** – globálna funkcia na obsluhu kliknutí na položky zoznamu, ktorá mení farbu položky a zaznamenáva čas pre konkrétnu úlohu.
- **window.startTimer** – globálna funkcia na spustenie časovača, ktorá tiež animuje ikonu časovača.
- **window.handleSendEmail** – funkcia na odoslanie emailu s časmi vykonania úloh, ktorá je volaná po dokončení všetkých úloh. Odosiela údaje na email cez službu EmailJS.
- **executeCustomCallback(value)** – Ak je v konfigurácii definovaný vlastný callback, odošle udalosť prostredníctvom `EventEmitter` s hodnotou (vykonanej akcie a získanej hodnoty, napríklad času) ako parametrom.

Poznámky

- Tento komponent sa používa ako interaktívny zoznam úloh v aplikáciách, kde používateľ môže sledovať čas strávený na každej úlohe.
- Emailové notifikácie sú odosielané cez službu EmailJS, ktorá umožňuje odoslanie výsledkov v textovej forme (CSV).
- Uloženie a načítanie počtu odoslaní a používateľského ID je zabezpečené prostredníctvom `LocalStorage`.

Komponent: MusicPlayer

Popis

Komponent `MusicPlayer` je určený na prehrávanie hudby a zobrazenie zoznamu piesní. Umožňuje používateľom prehľadávať piesne, prehrať vybranú pieseň a navigovať medzi rôznymi stránkami piesní. Tento komponent využíva knižnicu `A-Frame` na zobrazenie vizuálnych prvkov, ako sú názvy piesní, popisy a tlačidlá na prehrávanie hudby.

Použitie (React JSX)

```
<MusicPlayer configurationFile={musicConfigFile} />
```

Popis Parametrov

Názov	Typ	Popis
configurationFile	object	Konfiguračný súbor, ktorý obsahuje zoznam piesní. Tento súbor určuje názvy piesní, popisy, farby a URL adresy pre každú pieseň.

Použité časti A-Frame

- **a-entity** – hlavný kontajner pre celý prehrávač hudby, obsahujúci vizuálne prvky ako tlačidlá a textové položky.
- **a-xycontainer** – kontajner pre jednotlivé piesne a vizuálne prvky, ktorý zabezpečuje správne umiestnenie a organizáciu.
- **a-text** – používané na zobrazenie názvov a popisov piesní v zozname.
- **a-xyimage** – používané na zobrazenie interaktívnych tlačidiel na prehrávanie piesní.
- **a-sound** – komponent na prehrávanie zvuku v A-Frame.

Interné mechanizmy a funkcie

- **useState** – hook používaný na uchovávanie stavu, ako je zoznam piesní, aktuálna stránka a stav prehrávania piesne.
- **useEffect** – hook, ktorý načítava konfiguračný súbor a určuje zoznam piesní na základe poskytnutého súboru.
- **handlePlay** – funkcia, ktorá prehráva pieseň po kliknutí na tlačidlo pre prehrávanie. Ak je už nejaká pieseň prehrávaná, zastaví ju a spustí novú.
- **handlePrevious** a **handleNext** – funkcie na navigáciu medzi stránkami zoznamu piesní.
- **generateUniqueKey** – funkcia generujúca unikátny kľúč pre každý kontajner v zozname piesní.
- **getPageSongs** – funkcia na získanie piesní pre aktuálnu stránku, s ohľadom na maximálny počet piesní na jednej stránke.

- **executeCustomCallback(value)** – Ak je v konfigurácii definovaný vlastný callback, odošle udalosť prostredníctvom `EventEmitter` s hodnotou výberu ako parametrom.

Poznámky

- Piesne sú načítané z konfiguračného súboru, ktorý obsahuje rôzne informácie o piesňach ako názov, popis, farba a URL na zvukový súbor.
- Po kliknutí na tlačidlo pre prehrávanie sa zvuk začne prehrávať pomocou komponentu `a-sound`.

Komponent: WebSearch

Popis

Komponent `WebSearch` je určený na vytvorenie interaktívneho vyhľadávacieho poľa s vizuálnymi prvkami v aplikáciách využívajúcich `A-Frame`. Umožňuje používateľovi zadávať kľúčové slová do vyhľadávacieho poľa a kliknúť na tlačidlo pre vykonanie vyhľadávania. Tento komponent používa prispôsobené `A-Frame` komponenty pre zobrazenie vizuálnych prvkov ako tlačidlá a vstupné polia.

Použitie (React JSX)

```
<WebSearch />
```

Použité časti A-Frame

- **a-xycontainer** – kontajner pre celé vyhľadávacie pole, ktorý organizuje rôzne vizuálne prvky ako tlačidlá a vstupy.
- **a-xyimage** – používa sa na zobrazenie interaktívnych tlačidiel, ako je tlačidlo pre vykonanie vyhľadávania.
- **a-xyinput** – používa sa na zobrazenie vstupného poľa pre zadávanie textu (kľúčových slov) na vyhľadávania.
- **a-entity** – používa sa na obalovanie rôznych vizuálnych komponentov a ich správne umiestnenie v 3D priestore.
- **a-xykeyboard** – Virtuálna klávesnica na zadávanie textu.

Interné mechanizmy a funkcie

- `useEffect (listener)`: Pridá poslucháča na vlastnú udalosť `aframeSearchDataInput`. Keď používateľ zadá hľadaný výraz, vyvolá sa buď vlastná akcia z konfigurácie, alebo predvolené vyhľadávanie.
- `search(q)`: Predvolená funkcia – ukončí VR mód a otvorí Google vyhľadávanie s výrazom `q`.
- `executeCustomCallback(query)`: Ak je v konfigurácii definovaný vlastný callback, odošle udalosť cez `EventEmitter` s hľadaným výrazom ako parametrom.

Poznámky

- Pre správne fungovanie vyhľadávacej akcie je potrebné priradiť elementu, ktorý potvrdzuje akciu (napr. tlačidlu), atribút `id="input-button"`, keďže komponent A-Frame naslúcha udalostiam cez tento identifikátor a následne odošle vstup späť do Reactu.

Komponent: Paginator

Popis

Komponent `Paginator` je určený na navigáciu medzi stránkami alebo obsahom. Obsahuje dva vizuálne prvky – jeden pre pohyb na predchádzajúcu stránku (ľavý) a druhý pre pohyb na ďalšiu stránku (pravý). Tento komponent využíva A-Frame a prispôsobené komponenty na zobrazenie tlačidiel, ktoré reagujú na kliknutia a umožňujú používateľovi presúvať sa medzi stránkami alebo obsahom.

Použitie (React JSX)

```
<Paginator uniqueKey="left"
  onChange={handlePageChange} color="blue" />
<Paginator uniqueKey="right"
  onChange={handlePageChange} color="green" />
```

Popis Parametrov

Názov	Typ	Popis
uniqueKey	string	Unikátny kľúč pre identifikáciu a správne volanie funkcie (napr. <code>left</code> alebo <code>right</code>).
onChange	function	Funkcia, ktorá je zavolaná pri zmene stránky (napr. pri kliknutí na tlačidlo pre predchádzajúcu alebo ďalšiu stránku).
color	string	Farba tlačidla.
addBackground	boolean	Určuje, či sa má pridať pozadie okolo tlačidla. (default: <code>true</code>)
backgroundColor	string	Určuje farbu pozadia tlačidla (ak je <code>addBackground</code> nastavené na <code>true</code>).

Použité časti A-Frame

- **a-entity** – obal pre tlačidlo s nastavením pozície a farby pozadia.
- **a-xyimage** – zobrazuje interaktívne tlačidlo pre navigáciu (ľavé alebo pravé), ktoré reaguje na kliknutia.

Interné mechanizmy a funkcie

- **handlePrevious** – funkcia, ktorá je volaná pri kliknutí na tlačidlo na presun na predchádzajúcu stránku. Táto funkcia je spojená s globálnym objektom `window`, aby mohla byť dynamicky volaná podľa unikátneho kľúča.
- **window[handlePrevious\$uniqueKey]** – globálna funkcia, ktorá umožňuje reagovať na kliknutie na tlačidlo pre presun na predchádzajúcu stránku. (Zachytenie udalosti na strane A-frame)

Poznámky

- Indikujúce stav *disabled* je možné meniť interne v Reacte, až po komponenty v A-Frame.

Knižnica základných entít pre UI-menu

A-Frame Komponent: fixed-position

Popis

Komponent `fixed-position` slúži na ukotvenie A-Frame entity vo fixnej vzdialenosti pred kamerou. Je ideálny pre zobrazovanie prvkov používateľského rozhrania (napr. notifikačné okná, informačné panely), ktoré by mali byť vždy viditeľné pred používateľom bez ohľadu na jeho pohyb v priestore.

Použitie (HTML)

```
<a-entity fixed-position="distance: 0.15"></a-entity>
```

Parametre

Názov	Typ	Popis
distance	number	Určuje vzdialenosť entity od kamery v metroch. Predvolená hodnota je 0.1. Entity sa bude automaticky pohybovať tak, aby bola stále v tejto vzdialenosti.

Príklad s parametrami

```
<a-entity  
  geometry="primitive: plane; height: 0.3; width: 0.5"  
  fixed-position="distance: 0.2">  
</a-entity>
```

Dostupné metódy

- **calculatePosition(cameraPosition, cameraRotation)** - Interná metóda komponentu. Vypočíta pozíciu entity voči kamere na základe zadaných súradníc a rotácie kamery.

Udalosti a interakcie

Interakcia	Popis
keydown: x	Prepína viditeľnosť entity. Pri stlačení klávesy x sa komponent zobrazí alebo skryje a zároveň sa jeho rotácia nastaví podľa aktuálnej rotácie kamery.

Kontext v rámci UI systému

Komponent `fixed-position` je dôležitým nástrojom na tvorbu rozhraní, ktoré sa dynamicky prispôbujú pohybu používateľa v 3D prostredí. Umožňuje zachovať elementy používateľského rozhrania stále pred očami používateľa, čím zabezpečuje ich dostupnosť bez potreby manuálneho nasmerovania pohľadu. Je vhodný najmä pre zobrazovanie inštrukcií, dialógových okien alebo akýchkoľvek systémových notifikácií. V budúcnosti je možné komponent rozšíriť o podporu VR ovládačov (napr. `grip`, `bbutton`).

A-Frame Komponent: `b-button-listener`

Popis

Komponent `b-button-listener` slúži na spracovanie stlačenia tlačidla B na VR ovládači. Po stlačení tohto tlačidla sa prepína viditeľnosť určenej entity — konkrétne okna s ID `window`. Tento komponent je vhodný pre interaktívne rozhrania vo VR prostredí, kde je potrebné rýchlo zobrazíť alebo skryť menu alebo panel.

Použitie (HTML)

```
<a-entity b-button-listener></a-entity>
```

Príklad použitia

```
<a-entity
  id="controller"
  b-button-listener>
</a-entity>
```

```
<a-entity
  id="window"
  geometry="primitive: plane; height: 0.4; width: 0.6"
  visible="false">
</a-entity>
```

Dostupné metódy

- **toggleVisibility()** - Interná funkcia, ktorá prepne stav viditeľnosti entity s ID window, zároveň nastaví jej pozíciu a orientáciu podľa aktuálnej polohy a rotácie kamery.

Udalosti

Udalosť	Popis
bbuttondown	Udalosť vyvolaná stlačením tlačidla B na VR ovládači. Aktivuje funkciu <code>toggleVisibility</code> , ktorá zobrazí alebo skryje menu.

Kontext v rámci UI systému

Komponent `b-button-listener` je navrhnutý na použitie v prostredí VR, kde klasická interakcia cez klávesnicu alebo myš nie je možná. V spojení s komponentom `fixed-position` a elementom `window` umožňuje používateľom zobraziť alebo skryť používateľské rozhranie jednoducho stlačením tlačidla na ovládači. Tento spôsob ovládania výrazne zlepšuje použiteľnosť v imerzívnych prostrediach a podporuje plynulý zážitok bez potreby vizuálnych indikátorov alebo kurzora.

A-Frame Komponent: gui-interactable

Popis

Komponent `gui-interactable` zabezpečuje interaktívne správanie 3D prvkov v scéne. Umožňuje reagovať na kliknutia používateľa, pohyb myšou (`hover`), alebo na stlačenie kláves. Tento komponent slúži ako základný vstupný bod pre interakcie v rámci používateľského rozhrania (UI) nášho systému.

Použitie (HTML)

```
<a-entity
  gui-interactable="clickAction:
    navigate; hoverAction:
    highlight;">
</a-entity>
```

Parametre

Názov	Typ	Popis
<code>clickAction</code>	<code>string</code>	Akcia, ktorá sa vykoná pri kliknutí (napr. emitovanie udalosti, volanie funkcie).
<code>hoverAction</code>	<code>string</code>	Akcia, ktorá sa vykoná pri hoveri (pri prejdení myšou nad entitou).
<code>keyCode</code>	<code>number</code>	Voliteľný: špecifický kód klávesy, ktorý spustí kliknutie.
<code>key</code>	<code>string</code>	Voliteľný: znak klávesy, ktorý spustí kliknutie.

Príklad s parametrami

```
<a-entity
  geometry="primitive: box"
  gui-interactable="clickAction: openPanel; key: Enter">
</a-entity>
```

Dostupné metódy

Metóda	Vstup	Popis
setClickAction(action)	string	Dynamicky nastaví novú akciu, ktorá sa má vykonať pri kliknutí.

Udalosti

Udalosť	Popis
click	Emituje sa pri kliknutí myšou alebo stlačení priradenej klávesy. Umožňuje reakciu komponentov v scéne.

Kontext v rámci UI systému

Komponent `gui-interactable` je základným stavebným prvkom pri vytváraní interaktívnych elementov v používateľskom rozhraní. V spojení s ďalšími komponentmi slúži na vytvorenie navigačných prvkov, prepínačov, tlačidiel alebo polí s možnosťou výberu. V širšom kontexte umožňuje používateľovi ovládať celé rozhranie pomocou pohybov, pohľadov alebo klávesnice.

A-Frame Komponent: `xyimage`

Popis

Komponent `xyimage` predstavuje grafické interaktívne tlačidlo (napr. ikonu alebo obrázok) v 3D priestore s možnosťou animácie a interakcie. Využíva sa predovšetkým na tvorbu používateľského rozhrania. Komponent podporuje prispôsobenie vizuálnych vlastností, reakcie na udalosti (`hover`, `click`) a tiež prepínateľný stav (`toggle`).

Použitie (HTML)

```
<a-entity xyimage="src: #img; width: 0.5; height: 0.5;"></a-entity>
```

Parametre

Názov	Typ	Popis
-------	-----	-------

color	string	Farba pozadia obrázku.
fixed	boolean	Zatiaľ nevyužívané. Rezervované pre polohovanie.
alphaTest	number	Prah priehľadnosti pri rendrovaní obrázkov s alfa kanálom.
src	asset	Cesta k textúre alebo obrázku.
width	number	Šírka obrázka.
height	number	Výška obrázka.
position	string	Pozícia v priestore.
guiInteractable	string	Vstup pre prepojenie s komponentom gui-interactable.
clickChangeColor	string	Nepoužívané; plánované pre zmenu farby cieľového prvku.
on	string	Udalosť, na ktorú komponent reaguje (napr. click).
value	string	Popisná hodnota tlačidla (napr. pre WAI-ARIA).
fontSize	number	Veľkosť písma (nepoužíva sa priamo v tomto komponente).
fontFamily	string	Typ písma (rezerva pre rozšírenie).
fontColor	string	Farba textu (v prípade kombinácie s textovým prvkom).
borderColor	string	Farba orámovania (nepoužívaná).
focusColor	string	Farba pri zameraní.
backgroundColor	string	Farba pozadia obrázku.
hoverColor	string	Farba pri hover efekte (alebo zväčšenie).
activeColor	string	Farba pri aktivácii (napr. kliknutí).
toggle	boolean	Ak je true, komponent funguje ako prepínač.
toggleState	boolean	Aktuálny stav prepínača.

Príklad použitia

```
<a-entity
  xyimage="src: #icon;
    width: 0.6;
```

```

        height: 0.6;
        on: click;
        toggle: true;">
</a-entity>

```

Vnútorne funkcie a správanie

- **mouseenter** – pri prejení kurzorom zväčší ikonu, ak nie je v toggle móde.
- **mouseleave** – vráti ikonu do pôvodnej veľkosti a odstráni animácie.
- **click (resp. podľa atribútu on)** – buď prehrá klikáciu animáciu, alebo (ak `toggle = true`) prepne stav komponentu.
- **setActiveState(activeState)** – nastaví stav prepínača (aktívny / neaktívny) a upraví farbu pozadia podľa stavu.

Súvisiace komponenty

- **gui-interactable** – komponent zabezpečujúci interakciu cez atribúty `onclick`, `onhover`, `key-code`, ktoré sa mapujú cez primitívu `a-xyimage`. Bez tejto väzby nebude kliknutie vyvolávať logiku viazanú na externú funkciu.
- **gui-button** – voliteľný komponent, ktorého metóda `setActiveState()` je volaná pri toggle móde.

Primitíva: a-xyimage

```

<a-xyimage
  src="#img"
  onclick="myFunction"
  width="1"
  height="1"
  background-color="#333">
</a-xyimage>

```

Poznámka: Mappings v primitíve zabezpečujú automatické napojenie vlastností na komponenty `gui-interactable` a `xyimage`.

A-Frame Komponent: menu-icon

Popis

Komponent `menu-icon` predstavuje grafické tlačidlo (napr. ikonu) v 3D prostredí A-Frame s možnosťou interakcie a animácie. Tento komponent je vhodný na vytváranie menu ikon v používateľskom rozhraní NDMVR. Podporuje prispôbobe-
nie vizuálnych vlastností, ako sú farby, veľkosti, fonty, a tiež reakcie na udalosti ako `click` a `hover`. Okrem toho umožňuje prepínanie medzi aktívnym a neaktívnym stavom.

Použitie (HTML)

```
<a-entity
  menu-icon="src: #icon;
  width: 1;
  height: 1;
  position: 0 -2.5 0.1;">
</a-entity>
```

Parametre

Názov	Typ	Popis
color	string	Farba pozadia ikony.
fixed	boolean	Určuje, či je ikona pevná alebo pohyblivá. Predvolene je <code>true</code> .
alphaTest	number	Prahová hodnota priesvitnosti pri renderovaní obrázkov s alfa kanálom.
src	asset	Cesta k obrázku, ktorý bude použitý ako ikona.
width	number	Šírka ikony.
height	number	Výška ikony.
position	string	Pozícia v 3D priestore.
radius	number	Veľkosť zaoblenia rohov pozadia ikony.
on	string	Udalosť, na ktorú komponent reaguje (napr. <code>click</code>).
value	string	Popisná hodnota tlačidla.

fontSize	number	Veľkosť písma (nepoužíva sa priamo v tomto komponente).
fontFamily	string	Typ písma (rezerva pre rozšírenie).
fontColor	string	Farba textu (v prípade kombinácie s textovým prvkom).
borderColor	string	Farba orámovania ikony.
focusColor	string	Farba pri zameraní (hover).
backgroundColor	string	Farba pozadia ikony.
hoverColor	string	Farba pri hover efekte (alebo zväčšení).
activeColor	string	Farba pri aktivácii (napr. kliknutí).
toggle	boolean	Ak je true, komponent funguje ako prepínač stavu.
toggleState	boolean	Aktuálny stav prepínača.

Príklad použitia

```
<a-entity
  menu-icon="src: #icon;
    width: 1;
    height: 1;
    position: 0 -2.5 0.1;
    on: click;
    toggle: true;">
</a-entity>
```

Vnútorne funkcie a správanie

- **mouseenter** – pri prejení kurzorom zväčší ikonu, ak nie je v toggle móde.
- **mouseleave** – vráti ikonu do pôvodnej veľkosti a odstráni animácie.
- **click (resp. podľa atribútu on)** – prehrá klikáciu animáciu, alebo (ak toggle = true) prepne stav komponentu.
- **setActiveState(activeState)** – nastaví stav prepínača (aktívny / neaktívny) a upraví farbu pozadia podľa stavu.

Súvisiace komponenty

- **gui-interactable** – komponent zabezpečujúci interakciu cez atribúty `onclick`, `onhover`, `key-code`, ktoré sa mapujú cez primitívu `a-menu-icon`. Bez tejto väzby nebude kliknutie vyvolávať logiku viazanú na externú funkciu.
- **gui-button** – voliteľný komponent, ktorého metóda `setActiveState()` je volaná pri toggle móde.

Primitíva: a-menu-icon

```
<a-menu-icon
  src="#icon"
  onclick="myFunction"
  width="1"
  height="1"
  background-color="#333">
</a-menu-icon>
```

Poznámka: Mappings v primitíve zabezpečujú automatické napojenie vlastností na komponenty `gui-interactable` a `menu-icon`.

A-Frame Komponent: highlight

Popis

Komponent `highlight` slúži na zvýraznenie tlačidiel v 3D prostredí A-Frame. Tento komponent reaguje na udalosti ako `mouseenter`, `mouseleave` a `click`, pričom umožňuje vizuálne zmeny na tlačidlách (napríklad zmena farby) a interakcie s pozadím. Pomocou tejto komponenty môžeme vytvoriť interaktívne tlačidlá, ktoré sa menia podľa stavu (napr. zvýraznenie pri prechode myšou a aktivácia pri kliknutí).

Použitie (HTML)

```
<a-entity highlight>
  <a-entity class="menu-button" slice9="color: black;"></a-entity>
  <a-entity class="menu-button" slice9="color: black;"></a-entity>
  <!-- Viac tlačidiel -->
</a-entity>
```

Príklad použitia

```
<a-entity highlight>
  <a-entity class="menu-button" slice9="color: black;"></a-entity>
  <a-entity class="menu-button" slice9="color: black;"></a-entity>
  <!-- Viac tlačidiel -->
</a-entity>
```

Vnútorne funkcie a správanie

- **onClick** – pri kliknutí na tlačidlo sa zmení jeho farba na #046de7, tlačidlo sa zväčší a pridá sa stav clicked.
- **onMouseEnter** – pri prejdení kurzorom nad tlačidlo sa jeho farba zmení na #a1bddd, ostatné tlačidlá zmenia farbu na black.
- **onMouseLeave** – pri opustení kurzora z tlačidla sa jeho farba vráti na black, pokiaľ nebolo kliknuté.
- **reset** – pri kliknutí na pozadie sa všetky tlačidlá vrátia do pôvodného stavu, farby sa resetujú a animácie sa obnovia.

Súvisiace komponenty

- **slice9** – komponent pre prácu s farbami tlačidiel (využíva sa na zmenu farby pri interakciách).
- **menu-button** – trieda pre tlačidlá, na ktoré sa komponent highlight aplikuje.

Primitíva: a-highlight

```
<a-highlight>
  <a-entity class="menu-button" slice9="color: black;"></a-entity>
  <a-entity class="menu-button" slice9="color: black;"></a-entity>
</a-highlight>
```

Poznámka: Komponent highlight funguje na všetkých objektoch, ktoré majú triedu menu-button. Prvky s touto triedou budú reagovať na interakcie (hover, click) definované v komponente.

A-Frame Komponent: rounded

Popis

Komponent `rounded` slúži na vytváranie 3D objektov s zaoblenými rohmi v A-Frame. Tento komponent umožňuje nastavenie veľkosti, farby, a priehľadnosti objektu. Umožňuje definovať špecifické rádiusy pre jednotlivé rohy (horný ľavý, horný pravý, dolný ľavý, dolný pravý), alebo použiť rovnaký rádius pre všetky rohy. Jeho výhodou je flexibilita v nastavení geometrie a vzhľadu zaoblených tvarov v 3D priestore.

Použitie (HTML)

```
<a-entity
  rounded="width: 2;
  height: 2;
  radius: 0.3;
  color: #FF0000;">
</a-entity>
```

Parametre

Názov	Typ	Popis
<code>enabled</code>	boolean	Určuje, či je komponent povolený. Predvolené nastavenie je <code>true</code> .
<code>width</code>	number	Šírka objektu. Predvolené nastavenie je 1.
<code>height</code>	number	Výška objektu. Predvolené nastavenie je 1.
<code>radius</code>	number	Všeobecný rádius pre všetky rohy. Predvolené nastavenie je 0.3.
<code>topLeftRadius</code>	number	Rádius pre horný ľavý roh. Predvolené nastavenie je -1, čo znamená, že použije hodnotu <code>radius</code> .
<code>topRightRadius</code>	number	Rádius pre horný pravý roh. Predvolené nastavenie je -1, čo znamená, že použije hodnotu <code>radius</code> .

bottomLeftRadius	number	Rádus pre dolný ľavý roh. Predvolené nastavenie je -1, čo znamená, že použije hodnotu radius.
bottomRightRadius	number	Rádus pre dolný pravý roh. Predvolené nastavenie je -1, čo znamená, že použije hodnotu radius.
depthWrite	boolean	Určuje, či sa má objekt zapisovať do hĺbkovej vrstvy (depth buffer). Predvolené nastavenie je true.
polygonOffset	boolean	Aktivuje offset pre polygóny, aby sa zabránilo prekrytiu s inými objektmi. Predvolené nastavenie je false.
polygonOffsetFactor	number	Nastavenie faktorov pre offset polygónov. Predvolené nastavenie je 0.
color	color	Farba objektu v hexadecimálnom formáte. Predvolené nastavenie je #F0F0F0.
opacity	number	Priehľadnosť objektu. Predvolené nastavenie je 1.

Príklad použitia

```
<a-entity
  rounded="width: 2;
    height: 2;
    radius: 0.3;
    color: #FF0000;">
</a-entity>
```

Vnútorne funkcie a správanie

- **init** – pri inicializácii komponenta vytvorí 3D objekt s nastavenými parametrami a priradí ho k entite.
- **update** – pri zmene parametrov komponent aktualizuje geometriu objektu, jeho farbu a priehľadnosť, ak je komponent povolený (enabled).
- **updateOpacity** – zaisťuje, že hodnota opacity je v rozmedzí od 0 do 1, a nastavuje transparentnosť objektu.

- **remove** – pri odstránení komponentu sa odstráni 3D objekt z entity.
- **draw** – vytvára geometriu zaobleného tvaru na základe zadáných parametrov a vytvára `THREE.ShapeGeometry`.

Súvisiace komponenty

- **THREE.ShapeGeometry** – geometria, ktorá sa využíva pri vytváraní zaoblených tvarov.
- **MeshStandardMaterial** – materiál pre 3D objekty v A-Frame, ktorý umožňuje nastavenie farby a ďalších vlastností.

A-Frame Komponent: gui-button-rounded

Popis

`gui-button-rounded` je tlačidlový komponent s oblými hranami určený pre použitie výlučne na strane A-Frame. Tento komponent poskytuje interaktívny dizajn s možnosťou vizuálneho zvýraznenia pri hoveri, kliknutí alebo fokuse a zároveň podporuje prepínací (toggle) režim.

Použitie (HTML primitíva)

```
<a-gui-button-rounded
  value="Start"
  onclick="startScene"
  font-color="#fff"
  background-color="#333"
  active-color="#f90"
  hover-color="#666"
  toggle="true">
</a-gui-button-rounded>
```

Parametre

Parameter	Typ	Popis
-----------	-----	-------

on	string	Typ udalosti, ktorá spúšťa akciu (napr. <code>click</code> , <code>hover</code>).
value	string	Text, ktorý sa zobrazí v tlačidle.
fontSize	number	Veľkosť písma.
fontFamily	string	Typ písma použitý na vykreslenie textu.
fontColor	string	Farba textu.
borderColor	string	Farba okraja komponentu (pri fokusovaní).
focusColor	string	Farba, ktorá sa použije pri získaní fokusu.
backgroundColor	string	Výchozia farba pozadia.
hoverColor	string	Farba pri hoveri myšou alebo pri zvýraznení.
activeColor	string	Farba pri aktivácii alebo kliknutí.
toggle	boolean	Ak <code>true</code> , tlačidlo funguje ako prepínač (stav ON/OFF).
toggleState	boolean	Počiatočný stav v toggle režime.
position	string	Pozícia v priestore (napr. " <code>0 1 0</code> ").

Metódy

Názov	Vstup	Popis
<code>setActiveState(state)</code>	boolean	Nastaví aktuálny stav prepínača (zmení vzhľad podľa <code>toggleState</code>).
<code>setText(text)</code>	string	Dynamicky aktualizuje text v tlačidle.

Udalosti

- **click / on** - Spustí definovanú akciu a prípadne animáciu. Môže byť aktivovaná myšou, pohľadom alebo klávesou.
- **on callback** - z premennej `guiInteractable.clickAction` sa získa názov funkcie, ktorá sa následne vyhladá v globálnom `window` objekte. Ak existuje a je typu `function`, zavolá sa s aktuálnou udalosťou `event` ako argumentom.
- **mouseenter / mouseleave** - Zmení vzhľad tlačidla (hover efekt).
- **focus / blur** - Umožňuje zvýrazniť komponent pomocou klávesnice alebo navigácie.

Sémantická podpora a prístupnosť

- Komponent má nastavený `role="button"` a `tabindex="0"` pre klávesnicovú navigáciu.
- Obsahuje `aria-label` pre asistívne technológie.

V kontexte UI systému

`gui-button-rounded` je vizuálnym a interakčným prvkom v rámci modulárnej knižnice používateľského rozhrania, založeného na A-Frame. V kombinácii s komponentami ako `gui-interactable`, `rounded` tvorí základné stavebné bloky tlačidlových panelov, navigačných menu, prepínačov režimov a interaktívnych HUD elementov.

Pomocou podpory pre rôzne vstupy (myš, klávesnica, pohľad, touch) je pripravený na nasadenie v širokom spektre zariadení — od desktopu po headsety.

A-Frame Komponent: `gui-button`

Popis

Komponent `gui-button` slúži na vytváranie interaktívnych tlačidiel v A-Frame. Tento komponent umožňuje nastavenie rôznych parametrov, ako sú farby, veľkosti, a text na tlačidlo. Komponent podporuje rôzne stavy tlačidla, ako sú normálny, hover, aktívny a focus stav, čím zaisťuje dynamickú interaktivitu.

Použitie (HTML)

```
<a-gui-button
  value="Kliknite ma"
  width="2"
  height="1"
  color="#00ff00"
  hover-color="#ff0000"
  active-color="#0000ff"
  onclick="function"
>
</a-gui-button>
```

Parametre

Názov	Typ	Popis
enabled	boolean	Určuje, či je komponent povolený. Predvolené nastavenie je <code>true</code> .
width	number	Šírka tlačidla. Predvolené nastavenie je 1.
height	number	Výška tlačidla. Predvolené nastavenie je 1.
color	color	Farba tlačidla v hexadecimálnom formáte. Predvolené nastavenie je <code>#F0F0F0</code> .
hover-color	color	Farba tlačidla pri prechode myšou. Predvolené nastavenie je <code>#B0B0B0</code> .
active-color	color	Farba tlačidla v aktívnom stave (po kliknutí). Predvolené nastavenie je <code>#B0B0B0</code> .
focus-color	color	Farba tlačidla pri získaní focusu. Predvolené nastavenie je <code>#FF6347</code> .
font-size	number	Veľkosť písma na tlačidle. Predvolené nastavenie je 0.2.
font-family	string	Písmo použité na tlačidle. Predvolené nastavenie je <code>Arial</code> .
font-color	color	Farba textu na tlačidle. Predvolené nastavenie je <code>FFFFFF</code> .
toggle	boolean	Určuje, či je tlačidlo prepnutelné. Predvolené nastavenie je <code>false</code> .
toggle-state	boolean	Určuje aktuálny stav prepnutia tlačidla. Predvolené nastavenie je <code>false</code> .

Vnútročné funkcie a správanie

- **init** – pri inicializácii komponenta vytvorí 3D tlačidlo s nastavenými parametrami a priradí ho k entite.
- **update** – pri zmene parametrov komponent aktualizuje vzhľad tlačidla, jeho text a farbu, ak je komponent povolený (`enabled`).
- **updateState** – zaisťuje, že prepnutý stav (`toggle-state`) je aktualizovaný a zobrazuje správny vizuálny stav.
- **remove** – pri odstránení komponentu sa odstráni tlačidlo z entity.

- **draw** – vykreslí tlačidlo na obrazovke na základe zadaných parametrov.
- **onclick callback** - z premennej `guiInteractable.clickAction` sa získa názov funkcie, ktorá sa následne vyhľadá v globálnom `window` objekte. Ak existuje a je typu `function`, zavolá sa s aktuálnou udalosťou `event` ako argumentom.

Súvisiace komponenty

- **THREE.MeshBasicMaterial** – materiál pre 3D objekty v A-Frame, ktorý umožňuje nastavenie farby a ďalších vlastností.
- **a-text** – komponent na pridávanie textu na 3D objekty, ktorý sa môže využiť na nastavenie textu na tlačidlách.

Primitíva: a-gui-button

```
<a-gui-button
  value="Kliknite ma"
  width="3"
  height="1"
  color="#FF6347"
  hover-color="#FF4500"
  font-size="0.25"
  font-color="#FFFFFF"
  onclick="function">
</a-gui-button>
```

Poznámka: Komponent `gui-button` sa využíva na vytváranie tlačidiel v prostredí A-Frame, ktoré môžu reagovať na rôzne interakcie používateľa ako kliknutie, prechod myšou a získanie focusu.

A-Frame Komponent: search-box

Popis

Komponent `search-box` poskytuje interaktívne vyhľadávacie pole v prostredí A-Frame. Umožňuje používateľovi zadať hľadaný výraz, ktorý následne odošle cez `CustomEvent` s názvom `aframeSearchDataInput`. Ak nie je definovaná vlastná

akcia v React-e, otvorí sa nové okno s výsledkami Google vyhľadávania. Komponent je určený na integráciu vstupov používateľa do externého systému (napr. React alebo webové služby).

Použitie (HTML)

```
<a-entity search-box>
  <a-xyinput placeholder="Zadajte hľadaný výraz"></a-xyinput>
  <a-button id="input-button" value="Hľadať"></a-button>
</a-entity>
```

Parametre

Názov	Typ	Popis
input-button	string	ID tlačidla, ktoré spúšťa vyhľadávanie. Musí byť nastavené na "input-button", aby bol správne identifikovaný komponentom.
a-xyinput	element	Vstupné pole, ktoré obsahuje hodnotu hľadaného výrazu.

Príklad použitia

```
<a-entity search-box>
  <a-xyinput placeholder="Zadajte hľadaný výraz"></a-xyinput>
  <a-button id="input-button" value="Hľadať"></a-button>
</a-entity>
```

Vnútorne funkcie a správanie

- **init** – Pri inicializácii komponent získa referencie na vstupné pole a tlačidlo pomocou `querySelector`. Následne priradí sledovanie stavu na:
 - kliknutie tlačidla (`click`),
 - stlačenie klávesy Enter vo vstupnom poli (`keydown`).
- **_emitData(value)** – Odošle `CustomEvent` s názvom `aframeSearchDataInput` a telom `value: <hodnota>`.

- `_search(value)` – Voliteľná interná funkcia pre defaultné Google vyhľadávanie. Ukončí VR mód a otvorí výsledky v prehliadači.

Súvisiace komponenty

- **a-xyinput** – Komponent pre zadávanie textu.
- **a-button** – Používaný ako spúšťač akcie vyhľadávania.
- **a-scene** – Slúži na ukončenie VR módu pred presmerovaním na Google.

Primitíva: search-box

```
<a-entity search-box>
  <a-xyinput placeholder="Zadajte hľadaný výraz"></a-xyinput>
  <a-button id="input-button" value="Hľadať"></a-button>
</a-entity>
```

Poznámka: Aby komponent správne fungoval, je nevyhnutné, aby tlačidlo obsahovalo atribút `id="input-button"`. Bez tohto ID nebude možné zachytiť udalosť kliknutia.

A-Frame Komponent: xycontainer

Popis

Komponent `xycontainer` slúži na vytváranie kontajnerov, ktoré usporiadajú svojich potomkov v 3D priestore. Tento komponent umožňuje nastaviť rôzne možnosti rozmiestnenia, ako je smer (vertikálny alebo horizontálny), zarovnávanie položiek a ich správanie pri zmene veľkosti. Môže byť použitý na vytváranie flexibilných a responzívnych rozložení s nastavením smeru a zarovňovania položiek.

Použitie (HTML)

```
<a-entity xycontainer>
  <a-entity xyitem>...</a-entity>
  <a-entity xyitem>...</a-entity>
</a-entity>
```

Parametre

Názov	Typ	Popis
spacing	number	Určuje medzeru medzi jednotlivými položkami v kontajneri. Predvolené nastavenie je 0.0.
padding	number	Určuje vnútorný okraj kontajnera. Predvolené nastavenie je 0.
reverse	boolean	Určuje, či sa má usporiadanie otočiť. Predvolené nastavenie je false.
wrap	string	Určuje, či sa položky majú zbalením (wrap) alebo nie. Predvolené nastavenie je nowrap.
direction	string	Určuje smer, v akom sa majú položky usporiadať. Možnosti sú none, row, column, vertical, horizontal. Predvolené nastavenie je column.
alignItems	string	Určuje zarovnávanie položiek v kontajneri. Možnosti sú none, center, start, end, baseline, stretch. Predvolené nastavenie je none.
justifyItems	string	Určuje zarovnávanie položiek pozdĺž hlavného osy. Možnosti sú center, start, end, space-between, space-around, stretch. Predvolené nastavenie je start.
alignContent	string	Určuje zarovnávanie obsahu v kontajneri. Možnosti sú none, start, end, center, stretch. Predvolené nastavenie je prázdne ().

Príklad použitia

```
<a-entity xycontainer
  direction="column"
  spacing="0.1"
  padding="0.2"
  align-items="center">
```

```

<a-entity xyitem>...</a-entity>
<a-entity xyitem>...</a-entity>
</a-entity>

```

Vnútorne funkcie a správanie

- **init** – pri inicializácii komponenta priradí poslucháče udalostí pre pridané a odstránené deti, ako aj pre zmenu veľkosti. To spôsobí, že sa vždy vykoná aktualizácia, keď sa niečo zmení.
- **update** – komponent aktualizuje usporiadanie detí v kontajneri podľa nastavených parametrov (smer, zarovnávanie, atď.).
- **layoutLine** – nastaví pozíciu pre deti v kontajneri podľa smeru, zarovnávanie, a ďalších nastavení.

Súvisiace komponenty

- **xyrect** – geometria, ktorá sa používa na určenie rozmerov kontajnera.
- **xyitem** – komponent, ktorý určuje správanie jednotlivých položiek v kontajneri.

Primitíva: a-xycontainer

```

<a-xycontainer
  direction="row"
  spacing="0.2"
  padding="0.1"
  align-items="start"
  justify-items="center">
  <a-entity xyitem>...</a-entity>
  <a-entity xyitem>...</a-entity>
</a-xycontainer>

```

Poznámka: Komponent xycontainer je flexibilný a umožňuje vytvárať rôzne usporiadania pomocou rôznych nastavení pre zarovnávanie a smer položiek.

A-Frame Komponent: xyitem

Popis

Komponent `xyitem` sa používa na definovanie položiek, ktoré sa nachádzajú v kontajneri `xycontainer`. Tento komponent umožňuje nastaviť rôzne možnosti správania položiek v rámci kontajnera, ako sú zarovnávanie, rast a zmršťovanie.

Použitie (HTML)

```
<a-entity xyitem align="center" grow="2" shrink="1">...</a-entity>
```

Parametre

Názov	Typ	Popis
<code>align</code>	<code>string</code>	Určuje zarovnávanie položky v rámci kontajnera. Možnosti sú <code>none</code> , <code>center</code> , <code>start</code> , <code>end</code> , <code>baseline</code> , <code>stretch</code> . Predvolené nastavenie je <code>none</code> .
<code>grow</code>	<code>number</code>	Určuje, ako veľmi položka rastie v rámci kontajnera. Predvolené nastavenie je 1.
<code>shrink</code>	<code>number</code>	Určuje, ako veľmi sa položka zmršťuje v rámci kontajnera. Predvolené nastavenie je 1.
<code>fixed</code>	<code>boolean</code>	Určuje, či je položka fixná a nebude sa prispôsobovať veľkosti kontajnera. Predvolené nastavenie je <code>false</code> .

Príklad použitia

```
<a-entity xyitem align="start" grow="2" shrink="1">...</a-entity>
```

Vnútorne funkcie a správanie

- **update** – pri zmene parametrov komponenta aktualizuje usporiadanie položky v rámci svojho kontajnera `xycontainer`.

Súvisiace komponenty

- **xycontainer** – komponent, ktorý obsahuje položky xyitem a určuje, ako sa položky usporiadajú.

Primitíva: a-xyitem

```
<a-xyitem align="center" grow="1" shrink="2">...</a-xyitem>
```

Poznámka: Komponent xyitem je určený na definovanie správania položiek v kontajneri xycontainer. Umožňuje nastavenie zarovnávanie, rastu a zmršťovania, ktoré ovplyvňujú usporiadanie položiek.

A-Frame Komponent: xykeyboard

Popis

Komponent xykeyboard sa používa na vytvorenie virtuálnej klávesnice v prostredí A-Frame. Tento komponent umožňuje zobrazovať rôzne rozloženia klávesníc (plná, numerická) a interagovať s ňou prostredníctvom používateľských vstupov. Klávesnica sa skladá z rôznych blokov, ktoré môžu byť prispôsobené podľa potrieb používateľa (napríklad pre textové vstupy, čísla alebo špeciálne znaky).

Použitie (HTML)

```
<a-entity xykeyboard ime="true" distance="1.0"></a-entity>
```

Parametre

Názov	Typ	Popis
ime	boolean	Určuje, či klávesnica podporuje vstup pre písanie v japonskom jazyku (IME). Predvolené nastavenie je false.
distance	number	Určuje vzdialenosť, na ktorej bude klávesnica zobrazená od používateľa. Predvolené nastavenie je 0.7.

Príklad použitia

```
<a-entity xykeyboard ime="true" distance="1.0"></a-entity>
```

Vnútorné funkcie a správanie

- **init** – inicializuje komponent, pridá udalosť `xykeyboard-request` na detekciu požiadaviek na zobrazenie klávesnice.
- **show** – zobrazuje klávesnicu podľa typu (napr. numerická, plná alebo iné rozloženie).
- **hide** – skryje klávesnicu z obrazovky.
- **updateSymbols** – aktualizuje symboly na klávesoch pri prechode medzi rôznymi režimami klávesnice (napríklad pre zmenu Shift).

Súvisiace komponenty

- **xycontainer** – komponent, ktorý môže obsahovať položky, ktoré sa zobrazujú na klávesnici ako tlačidlá.
- **a-xycontainer** – primitív, ktorý môže byť použitý na usporiadanie tlačidiel klávesnice.
- **xylabel** – komponent na zobrazenie textu na tlačidlách klávesnice.

Primitíva: a-xykeyboard

```
<a-xykeyboard ime="false" distance="1.2"></a-xykeyboard>
```

Poznámka: Komponent `xykeyboard` poskytuje rozhranie na tvorbu virtuálnej klávesnice v prostredí A-Frame, pričom podporuje interaktivitu s klávesmi cez udalosť `mousedown`.

A-Frame Komponent: xyinput

Popis

Komponent `xyinput` je určený na implementáciu interaktívneho vstupu textu v prostredí A-Frame. Tento komponent poskytuje možnosť zadávať text s rôz-

nými parametrami ako farba kurzora, pozadie, placeholder, a typ textu (napríklad heslo). Umožňuje zobrazíť kurzor a umožňuje interakciu s textom pomocou klávesnice.

Použitie (HTML)

```
<a-entity xyinput
  value="Hello"
  placeholder="Enter text">
</a-entity>
```

Parametre

Názov	Typ	Popis
value	string	Určuje hodnotu zadanú používateľom v textovom poli. Predvolené nastavenie je prázdny reťazec.
placeholder	string	Určuje text, ktorý sa zobrazí, ak je vstup prázdny. Predvolené nastavenie je prázdny reťazec.
caretColor	string	Určuje farbu kurzora. Predvolené nastavenie je #0088ff.
color	string	Určuje farbu textu. Predvolené nastavenie je black.
bgColor	string	Určuje farbu pozadia vstupného poľa. Predvolené nastavenie je white.

Príklad použitia

```
<a-entity xyinput
  value="Hello"
  type="password"
  placeholder="Enter password"></a-entity>
```

Vnútorne funkcie a správanie

- **init** – inicializuje komponent, nastaví počiatočné hodnoty a vytvorí kurzor.

- **update** – aktualizuje hodnotu a zobrazenie textového poľa podľa aktuálneho stavu.
- **updateCursor** – aktualizuje pozíciu kurzora podľa aktuálneho vstupu.
- **caretpos** – vypočíta pozíciu kurzora na základe zadávaného textu.

Súvisiace komponenty

- **xylabel** – komponent používaný na zobrazenie textového obsahu v rámci vstupného poľa.
- **xyrect** – komponent na definovanie veľkosti a tvaru vstupného poľa.
- **xykeyboard** – komponent, ktorý poskytuje virtuálnu klávesnicu pre zadávanie textu.

Primitíva: a-xyinput

```
<a-xyinput value="Enter value" placeholder="Type here"></a-xyinput>
```

Poznámka: Komponent `xyinput` umožňuje vytvárať interaktívne vstupné polia v prostredí A-Frame s rôznymi možnosťami prispôsobenia ako farba, placeholder a kurzor. Tento komponent podporuje aj interakciu s klávesnicou a umožňuje prácu s textom priamo v 3D prostredí.