

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

**Virtuálne výučbové prostredie pre skladanie
počítačov**

Bakalárska práca

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

Virtuálne výučbové prostredie pre skladanie počítačov

Bakalárska práca

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

Košice 2025

Michal Sakáč

Abstrakt v SJ

Táto bakalárska práca sa zamierava na vytvorenie nového virtuálneho výučbového prostredia do systému LirkisEduVePn. Virtuálne prostredie je určené pre výučbu skladania počítačov. Výučbové úlohy v prostredí, ale aj samotné prostredie, je možné upravovať zmenou scenára, ktorý je vo forme Petriho siete. Prostredie je zamerané pre žiakov strednej školy ako doplnková aktivita k výučbe informatiky. Vytvoreniu tohto prostredia predchádzala analýza v ktorej sme použili metodiku systematického prehľadu literatúry. Na základe analýzy sme uskutočnili implementáciu riešenia, využívajúcu softvérový rámec A-Frame a fyzikálnu knižnicu Ammo.js. Výsledná forma prostredia je kancelária, v ktorej sú umiestnené počítačové komponenty a vopred poskladaný počítač. Podporované sú úlohy zamerané na skladanie počítača a výmenu komponentov v počítači.

Kľúčové slová v SJ

skladanie počítačov, virtuálna realita, rozšírená realita, Petriho siete, A-Frame, výučbové scenáre, imerzívne učenie

Abstrakt v AJ

This bachelor's thesis focuses on creating a new virtual learning environment for the LirkisEduVePn system. The virtual environment is designed for teaching computer assembly. The learning tasks in the environment, as well as the environment itself, can be modified by changing the scenario, which is in the form of a Petri net. The environment is aimed at secondary school students as a complementary activity to the teaching of computer science. The creation of this environment was preceded by an analysis in which we used the systematic literature review methodology. Based on the analysis, we implemented a solution using the A-Frame framework and the Ammo.js physics library. The final form of the environment is an office where computer components and a pre-assembled computer are placed. Supported tasks focus on computer assembly and component replacement in a computer.

Kľúčové slová v AJ

computer assembly, virtual reality, extended reality, Petri nets, A-Frame, learning scenarios, immersive learning

Bibliografická citácia

SAKÁČ, Michal. *Virtuálne výučbové prostredie pre skladanie počítačov*. Košice: Technická univerzita v Košiciach, Fakulta elektrotechniky a informatiky, 2025. 58s.
Vedúci práce: Ing. Štefan Korečko, PhD.

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

ZADANIE
BAKALÁRSKEJ PRÁCE

Študijný odbor: **Informatika**

Študijný program: **Informatika**

Názov práce:

Virtuálne výučbové prostredie pre skladanie počítačov
Virtual Educational Environment for Personal Computer Assembly

Študent: **Michal Sakáč**

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

Školiace pracovisko: **Katedra počítačov a informatiky**

Konzultant práce:

Pracovisko konzultanta:

Pokyny na vypracovanie bakalárskej práce:

1. Oboznámiť sa so systémom virtuálnych výučbových prostredí so scenármi vo forme Petriho sietí, vytvorenom na domovskom pracovisku.
2. Analyzovať existujúce riešenia pre výučbu skladania počítačov a relevantné požiadavky z pedagogickej praxe.
3. Na základe analýzy navrhnúť a implementovať výučbové prostredie zamerané na skladanie stolného počítača, vrátane scenárov vo forme Petriho sietí.
4. Návrhové a implementačné práce koordinovať s ďalšími riešiteľmi prác na systéme virtuálnych výučbových prostredí.
5. Vypracovať dokumentáciu podľa pokynov vedúceho práce.

Jazyk, v ktorom sa práca vypracuje: slovenský

Termín pre odovzdanie práce: 23.05.2025

Dátum zadania bakalárskej práce: 31.10.2024




.....
prof. Ing. Liberios Vokorokos, PhD.
dekan fakulty

Čestné vyhlásenie

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

Podakovanie

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

Taktiež by som sa chcel poďakovať Súkromnej strednej odbornej škole ekonomickej KOŠICKÁ AKADÉMIA, menovite riaditeľovi PaedDr. Petrovi Ivanovi, Ing. Adriane Popovičovej a Mgr. Michaela Haluškovej za to, že nám poskytnú možnosť testovať naše prostredie na ich škole a pomohli nám vybrať zameranie nášho prostredia.

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

Obsah

1	Úvod	1
2	Analýza súčasného stavu projektu	3
2.1	Práca Adama Kašelu	3
2.2	Práca Dmytra Demianenka a Lukáša Pisarčíka	4
2.3	Práca Juraja Rudyho	5
2.4	Spolupráca s Kristiánom Zsigóm	6
2.5	Záver	7
3	Systematický prehľad literatúry použitia virtuálnych prostredí na výučbu informatiky	9
3.1	Formulácia výskumnej otázky	9
3.2	Metodika získavania dát a ich spracovanie	10
3.3	Analýza výsledkov	11
3.4	Zhrnutie prác	12
3.4.1	Využitie virtuálnej reality	12
3.4.2	Využitie obohatenej reality	13
3.4.3	Počítačové aplikácie a hry	14
3.4.4	Spojenie videohier a výučby	16
3.5	Záver systematického prieskumu	17
4	Návrh z pohľadu používateľa	19
4.1	3D modely	19
4.1.1	Úpravy 3D modelov	21
4.2	Scéna	24
4.3	Scenáre	30
5	Implementácia	34
5.1	Kolízie objektov	35
5.2	Spájanie objektov	36

5.3	Odpájanie objektov	38
5.4	Scenáre vo forme Petriho siete	39
5.5	Úpravy pre zlepšenie použiteľnosti	40
6	Vyhodnotenie	42
6.1	Príprava testovania	42
6.2	Priebeh testovania	46
6.3	Výsledky z dotazníkov	48
7	Záver	53
	Zoznam použitej literatúry	55
	Zoznam príloh	59
A	Celý model Petriho siete	60
B	Používateľská príručka	65
B.1	Funkcia programu	65
B.2	Inštalácia programu	65
B.2.1	Požiadavky na technické prostriedky	66
B.2.2	Spustenie projektu	66
B.3	Použitie programu	72
B.3.1	Prihlasovacia obrazovka	72
B.3.2	Hlavná obrazovka	73
B.3.3	Dashboard pre učiteľov a adminov	73
B.3.4	Virtuálna scéna	75
C	Systémová príručka	79
C.1	Funkcia programu	79
C.2	Popis programu	79
C.2.1	Popis riešenia	80
C.2.2	Štruktúra prostredia pre skladanie počítačov	80
C.3	Popis implementovaných súborov	81
C.4	Popis implementovaných A-Frame komponentov	83
C.5	Využitie technológie	85
C.6	Spustenie programu	85
C.7	Nasadenie na server	86

Zoznam obrázkov

3.1	Prienik vo výsledkoch v databázach Scopus a Web of Science	11
3.2	Distribúcia nájdených publikácií podľa roku	11
4.1	Pôvodný model kancelárskeho prostredia	23
4.2	Model kancelárskeho prostredia s odstránenými objektami	23
4.3	Model kancelárskeho prostredia s odstránenými tieňmi	24
4.4	Úloha na skladanie počítača	26
4.5	Úloha na výmenu komponentov	27
4.6	Uvítacia tabuľa	27
4.7	Zásluhy za 3D modely	28
4.8	Plagát s komponentami a ich názvami	28
4.9	Plagát s inštrukciami na spájanie komponentov	29
4.10	Model kancelárskeho prostredia vo finálnej scéne	29
4.11	Časť Petriho siete so scenárom pre skladanie počítača	30
4.12	Časť Petriho siete so scenárom pre výmenu komponentov	31
4.13	Časť Petriho siete so scenárom pre úspešné splnenie úloh	31
4.14	Časť Petriho siete so scenárom pre nápomocné plagáty	32
6.1	Graf znázorňujúci vek účastníkov	49
6.2	Graf znázorňujúci výsledky z dotazníka IPQ - poskytnutý nástro- jom IPQ Cal	51
B.1	Prihlasovacia obrazovka do pgAdmin	70
B.2	Okno so servermi v pgAdmin	70
B.3	Databáza v pgAdmin	70
B.4	Prihlasovacie okno	73
B.5	Hlavná obrazovka	73
B.6	Výber úlohy	74
B.7	Používateľský dashobard	75
B.8	Prehľad historie úloh pre skupinu	75

B.9	Prehľad historie úlohy jednotlivca	76
B.10	Odpojená grafická karta	77
B.11	Grafická karta pripojená k základnej doske	78

Zoznam tabuliek

3.1	Tabuľka výskumných otázok	10
3.2	Tabuľka dopytov hľadania v databázach	10
4.1	Tabuľka použitých 3D modelov	21
6.1	Tabuľka znázorňujúca stupnicu hodnotenia pre dotazník VRSQ . .	52
6.2	Tabuľka znázorňujúca výsledky z dotazníka VRSQ	52

Zoznam výpisov

4.1	Štruktúra súborov použitých assetov v prostredí	25
4.2	Štruktúra súborov scenárov	33
5.1	Štruktúra súborov použitých na vytvorenie prostredia pre sklada- nie počítačov	34
B.1	Príklad základného súboru docker-compose.yaml	67
B.2	Príklad súboru servers.json	68
B.3	SQL dotaz na pridanie používateľa s administrátorskými právami .	71
B.4	SQL dotazy na vytvorenie účtov pre testovanie	71
C.1	Štruktúra súborov použitých na vytvorenie prostredia pre sklada- nie počítačov	80

1 Úvod

Táto práca nadväzuje na projekt *LirkisEduVePn* vyvíjaný na našej katedre v rámci prác Adama Kašelu [1], Dmytra Demianenka [2], Lukáša Pisarčíka [3] a Jura-ja Rudyho [4]. Doterajší vývoj projektu podrobnejšie rozoberieme v kapitole 2. Projekt sa snaží o poskytnutie interaktívnejšej a imerzívnejšej formy výučby pre školy. Študenti sa môžu učiť vo virtuálnom prostredí pomocou počítača, mobilu ale aj rôznych zariadení pre virtuálnu a rozšírenú realitu. Zámer výučby vo virtuálnom prostredí je zvýšiť motiváciu študentov pre predmet výučby, aktívne zapájať všetkých študentov do aktivít, zvýšiť ich spokojnosť s výučbou a zlepšiť ich výsledky. Projekt už obsahuje funkcionality potrebné pre študentov na vykonávanie úloh vo virtuálnom prostredí a pre učiteľov na vytváranie scenárov výučby, prehľad o aktivitách študentov a ich výsledkoch.

V tejto práci rozšírime nateraz veľmi obmedzenú ponuku virtuálnych prostredí o ďalšie. V projekte máme dve virtuálne prostredia pričom obe sú vytvorené ako múzeum a sú zamerané na výučbu o dejepisu, konkrétne o Habsburskej monarchii. Ak chceme aby projekt bol použiteľný pre viacero školských predmetov, resp. učív, musíme zabezpečiť, aby boli dostupné rôzne virtuálne prostredia. Tvorba virtuálnych prostredí je náročná a zdĺhavá práca, ktorá pozostáva z nasledujúcich častí:

- 3D modelovania prostredia a objektov v prostredí - viac v sekcii 4.1,
- tvorby scenáru na báze Petriho siete - viac v sekcii 4.3,
- programovania interakcií medzi objektami v prostredí - viac v kapitole 5,
- testovania prostredia - viac v kapitole 6.

Kvôli náročnosti tvorby prostredí nebudeme v tejto práci schopní vytvoriť dostatočné množstvo virtuálnych prostredí pre všetky predmety. Zamerali sme sa predmet informatika a to konkrétne na výučbu o hardvéri počítača. Výber zamerania virtuálneho prostredia vyplýval z komunikácie so školou Súkromná stredná odborná škola ekonomická KOŠICKÁ AKADÉMIA, na ktorej budeme

prostredie testovať. Stretli sme sa s riaditeľom školy PaedDr. Petrom Ivanom, Ing. Adriánou Popovičovou a Mgr. Michaelou Haluškovou, ktorí nám pomohli vybrať zameranie prostredia. Preberali sme témy zamerané na cudzie jazyky, no v závere sme sa dohodli na informatike. Chceli sme dať žiakom možnosť vyskúšať si skladanie počítača, čo bolo v čase stretnutia žiakom nedostupné z dôvodu nedostatku vybavenia školy.

Formulácia úlohy

Prínosom tejto práce má byť nové virtuálne prostredie, zamerané na skladanie počítača, použiteľné pre projekt *LirkisEduVePn*, ktorý je vyvíjaný na našej katedre. Žiaci sa v tomto prostredí naučia identifikovať jednotlivé komponenty počítača, poskladajú si počítač a vyskúšajú si výmenu niektorého z komponentov.

Prvým krokom je analyzovať aktuálny stav projektu a zistiť jeho funkcionality. Potrebujeme zachovať všetky formy použiteľnosti pre naše a aj predošlé vytvorené prostredia. Tomuto kroku sa venuje kapitola 2.

Následne sa pozrieme na existujúce riešenia virtuálnych prostredí zameraných na skladanie počítača. Naším cieľom bude zistiť či takéto prostredie už existuje. Získame poznatky o tom, ako boli navrhnuté a aké technológie boli použité. Zistíme čo fungovalo a čo nie, aby sme sa vyhli chybám iných prác. Tento prieskum sme urobili v kapitole 3 metódou systematického prehľadu literatúry.

Potom sa pustíme do návrhu prostredia a jeho funkcionalít. Vyhľadáme si 3D modely potrebné pre naše prostredie a upravíme ich podľa našich potrieb. Z 3D modelov a doplnkových obrázkov vytvoríme 3D scénu, ktorá bude reprezentovať naše prostredie. Následne navrhujeme scenár pre našu scénu, ktorý bude obsahovať úlohy pre žiakov. Scenár bude vo forme Petriho siete, pričom budeme chcieť zachovať možnosť jednoduchšej úpravy scenárov pre učiteľov. Tento návrh je popísaný v kapitole 4.

S pripravenou 3D scénou a jasným scenárom sa pustíme do implementácie interakcií v prostredí. Zameriame sa hlavne na spájanie a odpájanie komponentov počítača. Naším cieľom bude, aby si žiaci mohli poskladať počítač a vyskúšať si výmenu komponentov. Celá implementácia je popísaná v kapitole 5.

Po dokončení implementácie sa pustíme do testovania prostredia. Dohodli sme sa so školou Súkromná stredná odborná škola ekonomická KOŠICKÁ AKADEMIA, že prostredie otestujeme na žiakoch 3. a 4. ročníka. Testovať budeme s použitím prilieb pre virtuálnu realitu, ale aj s počítačmi. Spôsob testovania je popísaný v kapitole 6 spolu s výsledkami testovania.

2 Analýza súčasného stavu projektu

V tejto kapitole sa pozrieme na vznik a vývoj projektu *LirkisEduVePn*, ktorý je predmetom tejto práce. Pozrieme na to, aké technológie a nástroje sa v projekte používajú a aké sú jeho hlavné časti. Zistíme aké funkcionality projekt ponúka, aby sme ich vedeli zachovať aj v našom prostredí.

2.1 Práca Adama Kašelu

Základy projektu *LirkisEduVePn* boli položené v roku 2022, keď Adam Kašela začal pracovať na svojej diplomovej práci [1]. Vytvoril virtuálne výučbové prostredie pre výučbu dejepisu, konkrétne zamerané na učivo o Habsburskej monarchii. Vedomosti, ktoré mali žiaci získať, boli čerpané z učebnice dejepisu pre 7. ročník základných škôl a 2. ročník gymnázia s osemročným štúdiom.

Prostredie bolo navrhnuté ako múzeum s exponátmi, ktoré žiaci mohli preskúmať a prečítať si o nich informácie. Obsahovalo 3 miestnosti, v ktorých bolo dokopy 15 exponátov a 3 úlohy pre žiakov. Žiaci si svoje vedomosti vedeli overiť formou kvízových otázok, na ktoré odpovedali interakciou s prostredím a to buď presunom predmetov na určené miesto alebo stlačením tlačidla na výber odpovedí.

Implementácia pozostávala z jednoduchého webového rozhrania s využitím softvérového rámca A-Frame pre podporu rozšírenej reality. Na scenár pre riadenie prostredia a úlohy pre žiakov bola použitá Petriho sieť. V tomto štádiu projektu ešte neexistoval systém pre učiteľov a žiakov, ktorý by sledoval ich aktivitu v prostredí. Projekt pozostával čisto z jedného prostredia a jedného scenára, ktorý bol pevne daný a neexistovala možnosť jeho úpravy.

Vyhodnotenie úspešnosti práce bolo vykonané prieskumom použiteľnosti, na ktorom sa zúčastnilo 29 respondentov, pričom 18 respondentov boli cieľovou skupinou. Väčšina respondentov mala predchádzajúcu skúsenosť s virtuálnou realitou, čo sa prejavilo na ich pozitívnom hodnotení projektu. Z výsledkov jeho prieskumu vieme jasne povedať, že žiaci majú záujem o interaktívnejšiu formu

výučby a virtuálne prostredie im môže pomôcť pri učení.

2.2 Práca Dmytra Demianenka a Lukáša Pisarčíka

Dmytro Demianenko [2] a Lukáš Pisarčík [3] spoločne pracovali na systéme okolo virtuálneho prostredia vytvoreného Adamom Kašlom [1]. Dmytro sa zameriaval na serverovú časť systému pre správu scén a scénarov. Lukáš pracoval na rozšírení virtuálneho prostredia s historickou expozíciou a komunikáciu medzi prostredím a serverom. Spolu navrhli systém rozdelený na frontend, backend a databázu.

Frontend je implementovaný cez webový softvérový rámec Angular, ktorý je vhodný pre tvorbu dynamických a interaktívnych webových aplikácií. Zameriava sa na vykreslenie grafického používateľského rozhrania a interakciou s používateľom, pričom komunikuje s backendom cez HTTP požiadavky s údajmi vo formáte JSON. Primárne na ňom pracoval Dmytro.

Backend je implementovaný v jazyku Java s využitím softvérového rámca Spring Boot. Zameriava sa na spracovanie požiadaviek od frontendu, komunikáciu s databázou a spracovanie dát. Dmytro pracoval na časti backendu spojenú so správou scén, scénarov, úloh a štatistiky používateľov. Lukáš pracoval na časti backendu spojenú s ukladáním aktivity používateľov v prostredí.

Pre databázu bola zvolená relačná databáza PostgreSQL. Sú v nej uložené údaje aplikácie a backend s ňou komunikuje pomocou SQL dotazov.

Kvôli zabezpečeniu konzistentosti implementácie bez ohľadu na prostredie a operačný systém serveru kde bude aplikácia nasadená, bola zvolená Docker kontajnerizácia. Toto rozhodnutie pomohlo zjednodušiť proces nasadenia aplikácie.

Dmytro pracoval na autentifikácii a autorizácii používateľov, správe scénarov, scén, úloh a skupín a štatistikách používateľov. V štatistikách sa zaznamenáva sa čas strávený v prostredí, celková úspešnosť vykonania úloh a jednotlivé pokusy o odpálenie prechodov v Petriho sieti. Študenti môžu byť zaradení do viacerých skupín a skupine môže byť priradených viacero úloh. Učiteľ si môže pozrieť aj štatistiky jednotlivých skupín a porovnať ich úspešnosť.

Lukáš pridal možnosť ovládania cez VR prilbu a pridal podporu pre jazykové súbory, čo umožňuje vytvárať viacero jazykových mutácií prostredia. Virtuálne prostredie rozšíril o 4 nové miestnosti, ktoré obsahovali 22 nových exponátov a 5 nových úloh pre žiakov. Tieto úlohy sú stále zamerané na učivo o Habsburskej monarchii, aby sa zachovala konzistencia s pôvodným prostredím. Pridal aj ukla-

danie aktivity študentov v prostredí, čo sa využilo na možnosť vrátiť sa ku „rozohranému“ prostrediu a pokračovať v ňom, ale aj na získanie štatistík o aktivite študentov. Jazykové verzie prostredia boli implementované pomocou JSON súborov. Každý jazykový súbor obsahuje identifikátory použité v scéne a scenári a ku nim je text v jazyku daného súboru.

Ich práca priniesla systém s následovnými funkcionalitami:

- autentifikácia a autorizácia pre rozdelenie rolí učiteľa a študenta,
- pridávanie scenárov (úlohy prostredia), scén (3D prostredí) a úloh (prostredie s úlohami pre študentov),
- skupiny pre študentov a priradenie úloh skupinám,
- štatistiky úspešnosti študenta a skupín,
- možnosť ovládania prostredia cez VR prilbu,
- možnosť pridávania jazykových súborov pre lokalizáciu textov v prostredí,
- rozšírenie pôvodného virtuálneho prostredia.

2.3 Práca Juraja Rudyho

Posledné úpravy projektu vykonal Juraj Rudy vo svojej práci [4] zameranej na rozšírenie funkcionalít systému, technické úpravy, nasadenie na server a prieskum porozumenia Petriho sieťam.

V rámci jeho analýzy skúmal porozumenie Petriho sieťam neodbornou verejnosťou pomocou vlastného prieskumu. Respondentov rozdelil na technickú skupinu, ktorá má všeobecné technické znalosti, ale nemajú vedomosti o Petriho sieťach, a na netechnickú skupinu, ktorá nemá technické znalosti a ani vedomosti o Petriho sieťach. Technická skupina mala väčšiu úspešnosť a potrebovala menej času na prípravu a riešenie úloh. Po prieskume boli respondentom vysvetlené chyby, ktorých sa dopustili, veľa respondentov sa z chýb poučilo a boli presvedčení, že by danú chybu už neopakovali.

Projekt sa nasadil na virtuálny server poskytovaný od Technickej univerzity v Košiciach. Ako operačný systém bol zvolený Linux, konkrétne distribúcia Ubuntu, pre jeho natívnu podporu Docker kontajnerizácie. Ako úložisko Docker obrazov bol zvolený Docker Hub, ktorý je štandardným úložiskom pre Docker obrazy.

Medzi technické úpravy patrilo pridanie responzivity frontendu, konkrétne navigačného menu a ovladacieho panelu. Pridal sa prehľad scenárov a možnosť odstránenia scenára. Doplnili sa informácie do *about* stránky. Upravila sa farbu textu, zarovnanie a medzery informačnej steny v scéne múzea, aby bola čitateľnejšia. Opravila sa počiatočná poloha jedného exponátu, ktorý levitoval nad podstavcom, a opravil sa miznúci text na menovkách exponátov.

Pridala sa možnosť ovládania pre mobilné zariadenia. Využívajú sa v nej senzory zariadenia pre dosiahnutie podobného zážitku ako pri použití VR prilby. Pohyb po prostredí sa ovláda naklonením zariadenia a rotácia v scéne sa ovláda rotáciou zariadenia okolo zvislej osi. Interakcia s objektami sa vykonáva dotykcom obrazovky zariadenia.

Ovládanie cez VR prilbu bolo rozšírené teleportáciu, ktorá pomáha s presunom na veľké vzdialenosti v prostredí. Taktiež sa opravilo presúvanie objektov pomocou ovládačov, ktoré bolo nesprávne implementované.

Pre zabezpečenie komunikácie medzi používateľovým zariadením a systémom bol pridán SSL certifikát, ktorý zabezpečuje šifrovanú komunikáciu cez HTTPS protokol. Popri tom sa upravila verzia HTTP protokolu na HTTP/2, ktorý je rýchlejší a efektívnejší. Zaviedli sa pravidlá pre vytváranie hesiel, v záujme zvýšiť bezpečnosť používateľov.

2.4 Spolupráca s Kristiánom Zsigóm

Kristián Zsigó [5] pracoval súčasne s nami na projekte. Jeho cieľom bolo implementovať sadu nových komponentov s účelom zjednodušiť tvorbu nových scén. Komponenty, ktoré vytvoril, zakomponoval do nového ukážkového prostredia, ktoré bolo pridané do systému. Toto prostredie nebolo zamerané na žiadne učivo, ale malo slúžiť ako ukážka pre učiteľov a tvorcov nových scén, aby si vedeli predstaviť, aké sú možnosti interakcií vo virtuálnom prostredí.

V prostredí ukázal nové aktivity, napríklad hádzanie lopty alebo usporiadanie a triedenie predmetov. Pridal možnosť oddelovania časti prostredia pomocou dverí, ktoré sa môžu otvárať pomocou tlačidla, páky alebo zadaním hesla na virtuálnej klávesnici. Okrem tvorby nových komponentov sa zameral aj na úpravu starých komponentov, napríklad komponentov pre komunikáciu s Petriho sieťou.

Keďže sme na našej scéne pracovali súčasne ako sa tieto nové komponenty vytvárali, tak sme nevyužili niektoré komponenty, ktoré mohli byť pre nás užitočné. Ale na základe jeho práce sme sa rozhodli využiť fyzikálnu knižnicu *Ammo.js* pre fyziku nášho prostredia. Taktiež sme využili komponenty a rozhrania, ktoré

upravil, pre Petriho sieť.

2.5 Záver

Projekt pozostáva z frontendu vytvorenom cez Angular spolu s využitím softvérového rámca A-Frame, backendu vytvoreného v jazyku Java s využitím softvérového rámca Spring Boot a databázy PostgreSQL. Obsahuje:

- autentifikáciu a autorizáciu pre rozdelenie rolí študenta, učiteľa a administrátora,
- správu scenárov, pozostávajúcich z Petriho siete a jazykových súborov,
- registráciu scén, čo sú 3D prostredia,
- správu úloh, ktoré sú zložené zo scenára a scény,
- správu skupín študentov a priradenie úloh skupinám,
- zaznamenávanie aktivity študentov pri riešení úloh a štatistiky úspešnosti študentov a skupín,
- 2 predpripravené scenáre a scény,
- 3 možnosti ovládania prostredia - cez počítač, cez VR prilbu a cez mobilné zariadenie.

Pre naše prostredie potrebujeme zachovať možnosť úpravy scenárov, a to možnosťou zmeniť Petriho sieť pre zmenu úloh alebo úpravu scény a možnosťou pridať jazykové súbory pre lokalizáciu textov v prostredí.

Spomínané predpripravené scenáre a scény sú obe zamerané na učivo o Habsburskej monarchii. Pôvodný scenár a scéna boli vytvorené Adamom Kašlom [1] a druhá verzia je rozšírenie vytvorené Lukášom Pisarčíkom [3]. Vytvorenie nových scenárov a scén je náročný a zdĺhavý proces, preto treba predpripraviť scenáre a scény pre rôzne predmety výučby.

Prieskum vykonaný v práci Adama Kašlu [1] ukazuje, že žiaci majú záujem o interaktívnejšiu formu výučby a virtuálne prostredie im môže pomôcť pri učení. Väčšina respondentov verila v úspešnosť virtuálneho prostredia ako modernú formu výučby na školách.

Výsledky prieskumu Juraja Rudyho [4] ukázali, že aj netechnická skupina učiteľov dokáže pochopiť Petriho sieť a využiť ich pri vytváraní scenárov pre virtuálne prostredie. To znamená, že ak učitelia budú mať k dispozícii scény vhodné

pre ich predmet vyučovania, tak budú schopní pre žiakov vytvoriť interaktívne prostredie, ktoré im pomôže pri učení. Preto má vývoj tohto projektu zmysel a po vytvorení dostatočného množstva scén môže byť vhodným doplnkom výučby na školách.

3 Systematický prehľad literatúry použitia virtuálnych prostredí na výučbu informatiky

V tejto kapitole sa pozrieme na výskumy, ktoré sa zaoberali využitím virtuálnych prostredí na výučbu skladania počítača. Skladanie počítača sme vybrali ako tému nášho virtuálneho prostredia na základe komunikácie so školou, na ktorej budeme prostredie testovať a vyhodnocovať. Skladanie počítača je súčasť informatiky, ktorá je u nás povinným predmetom [6] na základných školách a stredných školách a aj v ostatných krajinách sú snahy o zavedenie informatiky do školského vzdelávania, lebo počítačová gramotnosť je zručnosť, ktorú každé dieťa potrebuje rozvíjať [7]. Na školách s minimálnym počítačovým vybavením je ťažké poskytnúť študentom praktické skúsenosti so skladaním počítača, preto väčšinou získajú iba teoretické vedomosti o počítačových komponentoch a ich funkciách. Naše prostredie môže takýmto školám poskytnúť praktickú formu výučby skladania počítača.

Virtuálne prostredia, ktoré je možné ovládať s VR prílami, benefitujú z možnosti interakcie s prostredím, preto by malo byť zamerané na fyzické objekty, v našom prípade napríklad na hardvér počítača.

V tejto kapitole označujeme rôzne formy virtuálnych prostredí ako hry. Nie všetky virtuálne prostredia sú hry v pravom slova zmysle, ale kvôli ich podobnosti, využitiu herných prvkov, herných vývojových prostredí a iných herných elementov ich označujeme ako hry. Je dôležité myslieť na to, že ich zameranie je vzdelávanie a nie zábava.

3.1 Formulácia výskumnej otázky

Výskumnú otázku zameriavame na virtuálne prostredia na skladanie počítača. Našou motíváciou je nájsť údaje o virtuálnych prostrediach s rovnakým zamera-

Výskumná otázka	Motivácia
Existujú virtuálne prostredia zamerané na výučbu o hardvéri počítača?	Nájsť údaje o virtuálnych prostrediach zameraných na hardvér počítača.

Tabuľka 3.1: Tabuľka výskumných otázok

Databáza	Dopyt na databázu
Scopus	TITLE-ABS-KEY (("education" OR "educational" OR "learning" OR "teaching") AND ("computer assembly" OR "computer hardware architecture" OR "hardware installation" OR "hardware replacement") AND ("game" OR "simulation" OR "virtual environment" OR "virtual scene" OR "virtual world"))
Web of Science	TS=(("education" OR "educational" OR "learning" OR "teaching") AND ("computer assembly" OR "computer hardware architecture" OR "hardware installation" OR "hardware replacement") AND ("game" OR "simulation" OR "virtual environment" OR "virtual scene" OR "virtual world"))

Tabuľka 3.2: Tabuľka dopytov hľadania v databázach

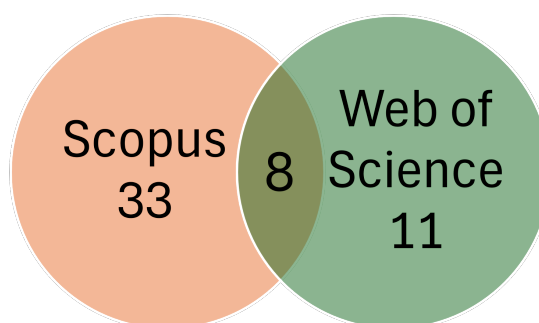
ním, ktoré budeme môcť využiť pri vývoji nášho virtuálneho prostredia ako inšpiráciu. Tabuľka 3.1 obsahuje formulované otázky, spolu s motiváciou prečo hľadáme odpovede v literatúre.

3.2 Metodika získavania dát a ich spracovanie

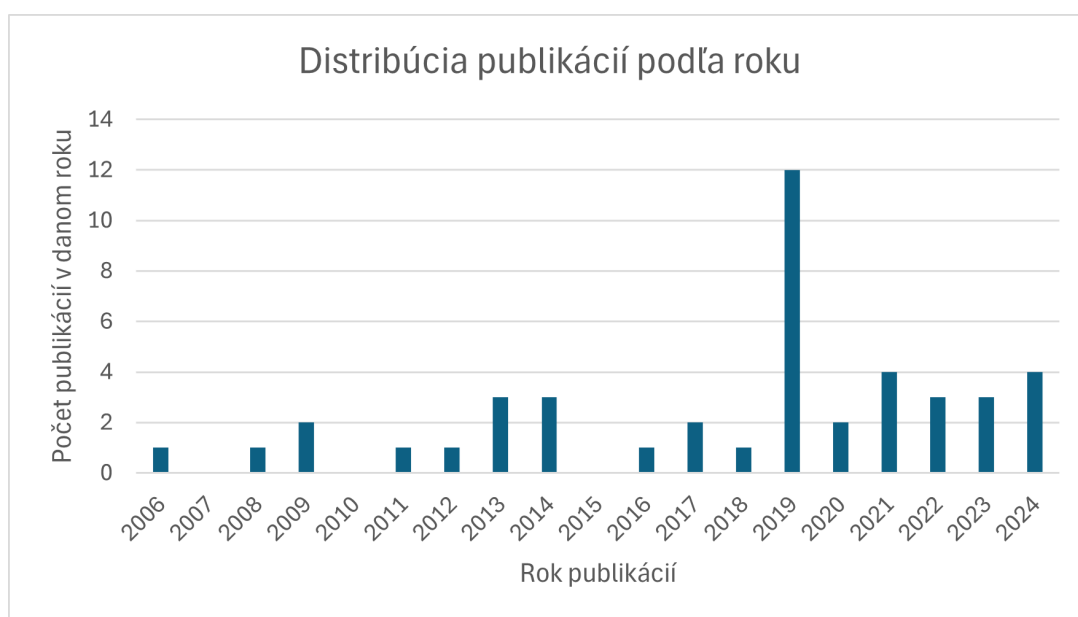
Po formulácii výskumnej otázky si musíme zadať spôsob získavania dát. To vykonáme dopytmi na databázy vedeckých štúdií. Rozhodli sme sa dopytovať v nasledujúcich databázach:

- **Scopus** - rozsiahla multidisciplinárna databáza abstraktov a citácií recenzovanej literatúry.
- **Web of Science** - medzinárodná databáza vedeckých citácií a abstraktov.

Tabuľka 3.2 obsahuje dopyty na databázy, ktoré sme použili na získanie vedeckých štúdií. Po vyhľadávaní v databázach sme získali 33 výsledkov z databázy



Obrázok 3.1: Prienik vo výsledkoch v databázach Scopus a Web of Science



Obrázok 3.2: Distribúcia nájdených publikácií podľa roku

Scopus a 11 výsledkov z databázy Web of Science. Pomocou aplikácie Microsoft Excel sme odstránili 8 duplicitných výsledkov získaných z databázy Web of Science. Na obrázku 3.1 je znázornený prienik výsledkov z databáz Scopus a Web of Science. Celkovo sme získali 36 unikátnych výsledkov.

3.3 Analýza výsledkov

V tejto kapitole analyzujeme získané výsledky z databáz Scopus a Web of Science, ktoré už máme vytriedené od duplicitných výsledkov. Graf na obrázku 3.2 zobrazuje distribúciu nájdených publikácií podľa roku. V roku 2019 vidíme veľký nárast publikácií, hlavne v databáze Scopus. Tento nárast je spôsobený väčším množstvom prehľadov o 4. ročníku konferencie aplikovanej vedy a inžinierstva, ako je napríklad [8]. Tieto prehľady mali v abstrakte zoznam tém prezentovaných na konferencii, medzi ktorými je aj téma implementácie algoritmu *depth-first*

search na výučbových 3D simuláciach zameraných na skladanie počítača.

3.4 Zhrnutie prác

Po preštudovaní získaných prác sme identifikovali relevantné práce, ktoré sú zaujímavé z hľadiska nášho výskumu. Cieľom všetkých prác bolo poskytnúť praktickú formu vzdelávania o skladaní počítača pre určitú skupinu ľudí, ktorý nemajú prístup ku skutočnému hardvéru. Našli sme práce využívajúce rôzne formy interakcie s virtuálnym prostredím, ako napríklad zariadenia pre virtuálne a rozšírené reality alebo počítačové hry. Kvôli množstvu prác sme tieto práce sme rozdelili podľa typu interakčného prostredia, ktoré používali.

Pri zlepšovaní dopytov na databázy sme našli jednu prácu [9], ktorá sa pozera na spojenie videohier a výučby v školskom prostredí. Nepopisuje žiadnu konkrétnu aplikáciu alebo hru, ale skôr sa zaoberá problémami a výzvami herného vzdelávania. Kvôli relevantnosti problémov z tejto práce sme sa rozhodli zahrnúť ju do nášho prehľadu literatúry, aj keď nie je súčasťou výsledkov z našich dopytov. Zamerali sme sa na ňu v podsekcii 3.4.4.

3.4.1 Využitie virtuálnej reality

Práce popísané v tejto sekcii sa zaoberajú využitím virtuálnej reality na výučbu o skladaní počítača. Tieto práce využívajú VR prilby a ovládače alebo sledovanie rúk cez kamery a senzory VR prilby na interakciu s virtuálnym prostredím.

Prostredie z práce [10] bolo modelované ako škola, kde bola chodba ako začiatočná oblasť a učebňa ako miesto pre skladanie počítača. Na chodbe bol návod na ovládanie hry, interakciu s objektmi a informácie o hre. V učebni bola tabuľa, na ktorej sa zobrazovali inštrukcie pre skladanie počítača a informácie o počítačových komponentoch. Počítač sa skladal na stole, od ktorého bol dobrý výhľad na tabuľu. Hráča v prostredí sprevádza virtuálny robotický asistent, ktorý poskytuje návody a rady.

V spomenutej práci použili herné vývojové prostredie Unreal Engine pre jeho nástroje na tvorbu fotorealistických prostredí a jednoduchému blokovému programovaniu. Mnohé objekty použité v prostredí boli stiahnuté priamo z Unreal Engine Marketplace, čo je obchod s obsahom pre toto herné vývojové prostredie. Niektoré objekty a ich variácie boli upravené alebo vytvorené v 3D modelovacom programe Blender.

Porovnali sa tu 3 formy výučby a to bežná online prednáška, hra pre počítač ovládaná klávesnicou a myšou a hra pre virtuálnu realitu ovládaná ovládač-

mi a VR prilbou. Výsledky ukázali, že hra pre virtuálnu realitu mala pozitívny vplyv na spokojnosť študentov a malý vplyv na ich výsledky, hlavne na vizuálne znalosti.

Textové inštrukcie nemusia byť pre všetkých študentov dostatočné, preto autori tejto práce [11] použili videá animácií na vysvetlenie postupu skladania počítača. Tieto animácie sa prehrávajú priamo v prostredí a sú dostupné počas tréningovej fázy. Pomáhajú študentom pochopiť postup skladania počítača pri možnosti hrania s klávesnicou a myšou. Keď sa používatelia rozhodnú skúsiť skladanie počítača, tak dostanú inštrukciu a majú ju vykonať bez pomoci. Ak ju nevzdĺadnu vykonať, tak sa môžu vrátiť k animácii a pozrieť si ju znova. Je to dobrý spôsob pre výučbu bez vysvetlenia teórie pred praxou, no je to určite náročnejšie na implementáciu do virtuálneho prostredia.

V nasledujúcej práci [12] experimentovali so sledovaním rúk pre virtuálne, rozšírené a zmiešane reality. Nás zaujalo prostredie pre skladanie počítača, ktoré bolo vytvorené pre výhradne pre virtuálnu realitu. Presnosť sledovania rúk sa zhoršila pri slabom osvetlení. Niektorí používatelia sa sťažovali, že nedostatok hmatového spätného väzby znižuje pohlcujúci zážitok z prostredia. Prostredie skladania počítača obsahovalo aj realistickú ľudskú postavu, ktorá bola ovládaná veľkým jazykovým modelom. Táto postava dokázala reagovať na akcie používateľa ako napríklad chytenie počítačového komponentu a vedele poskytnúť inštrukcie na jeho inštaláciu.

Namiesto simulovaného pomocníka použili v práci [13] kooperatívnu formu výučby. Dvaja používatelia, obidvaja s VR prilbami, si mohli navzájom pomáhať pri skladaní počítača. Vedia si posúvať objekty a komunikovať medzi sebou. Komunikácia cez internet bola limitovaná, lebo hlasové nahrávanie sa transformovalo na text a ten sa zobrazí druhej osobe v prostredí cez robotického pomocníka. Tento prístup bol určite jednoduchší na implementáciu ako hlasová komunikácia v reálnom čase, no mohol byť frustrujúci pre používateľov.

3.4.2 Využitie obohatenej reality

V tejto sekcii sa pozrieme na práce, ktoré využívali obohatenú realitu na výučbu o skladaní počítača. Táto forma reality kombinuje reálny svet ako primárne prostredie s virtuálnymi objektmi namapovanými na pozíciu reálnych objektov.

Pomocou princípov *marker-based augmented reality* autori v práci [14] vytvorili aplikáciu na skladanie počítača. Aplikácia bežala na mobile, pričom fotoaparátom snímala QR kód, resp. značku, ktorá bola umiestnená na stole. Na základe polohy a orientácie značky sa zobrazovali počítačové komponenty na správnej

pozícií. Samotné skladanie prebiehalo s využitím dotykovej obrazovky mobilu.

Testovanie aplikácie ukázalo, že aplikácia nebola intuitívna. Náš predpoklad je, že to môže byť spôsobené ovládaním mobilom a nedostatkom inštrukcií pre používateľov. V našom virtuálnom prostredí veríme, že použitie virtuálnej prilby s ovládačmi bude intuitívnejšie a pohodlnejšie pre používateľov. Taktiež sa zameriame na poskytnutie inštrukcií pre používateľov, ktoré budú dostupné v rámci prostredia.

V nasledujúcej práci [15] sa zameriavali na využitie špecialneho ovládača, *Leap Motion Controller*, na sledovanie rúk. Vytvorili virtuálne prostredie hrateľné pomocou virtuálnej aj rozšírenej reality a aj pomocou klávesnice a myši. Na skladanie počítača sa využíval už skôr spomínaný ovládač, ktorý funguje ako zariadenie na sledovanie rúk. Je položené na stôl pred používateľom a sleduje pohyb rúk nad ním.

Interakcie boli na základe gest, ktoré tento ovládač vedel rozpoznať. Tento spôsob ovládania sa porovnal s klávesnicou a myšou a výsledky ukázali, že ovládanie gestami bolo intuitívnejšie a jednoduchšie. Chýba nám žiaľ porovnanie s ovládaním pomocou ovládačov virtuálnej reality, ktoré sú štandardnou súčasťou VR prilb, preto nevieme posúdiť, či by to bola vhodná alternatívna voľba ovládania pre naše virtuálne prostredie.

3.4.3 Počítačové aplikácie a hry

Práce v tejto sekcii nemajú podporu žiadnej formy rozšírenej reality. Sú ovládateľné iba na počítači s klávesnicou a myšou. Naše prostredie bude ovládateľné aj touto formou, preto sa nám zdajú tieto práce relevantné.

Simulátor vytvorený v práci [16] má lineárnu postupnosť úloh, študenti ich musia vykonať v určenom poradí. Prostredie simulátora je veľké a navádza na hľadanie objektov a náučných materiálov pohybom po prostredí. Nesprávne rozhodnutia sú penalizované a po 3 chybách musí študent začať úlohu odznova. Takýto prístup nepovažujeme za vhodný, preto v našom prostredí chceme umožniť študentom vykonávať úlohy v ľubovoľnom poradí a bez akejkoľvek penalizácie za nesprávne rozhodnutia, lebo môže viesť k frustrácii študentov. V našom prostredí chceme umožniť študentom voľnú interakciu s objektmi a v prípade nesprávnych rozhodnutí ich chceme naviesť k správnejmu riešeniu problému. Avšak je potrebné poznamenať, že spôsob plnenia úloh v našom prostredí závisí od scenára, ktorý môže byť upravený. Ak sa učiteľ, alebo iný tvorca scenáru, rozhodne, že chce mať úlohy v určitom poradí, tak môže navrhnúť scenár tak, aby sa úlohy plnili v určenom poradí.

Simulátor z predošlého odseku bol vyvorený pomocou herného vývojového prostredia Unity. Toto herné vývojové prostredie použili aj autori práce [17]. Ich počítačové komponenty boli fotorealistické, ale ich prostredie bolo prázdne. Zamerali sa čisto na skladanie počítača a nie na vytvorenie vizuálne záživného prostredia. Na rozdiel od iných prác, v spomenutej práci bolo možné aj rozobratie počítača.

Herné vývojové prostredia pomáhajú pri tvorbe virtuálnych prostredí, no nie sú nevyhnutné. Našli sme aj práce [18] a [19], ktoré chceli vytvoriť virtuálne prostredie pre webový prehliadač. Obe práce využili modelovací jazyk VRML, pomocou ktorého pripravili súbory pre zobrazenie scény s 3D objektmi priamo v prehliadači. Prehliadač pre tieto prostredia nevedel plne využiť výkon počítača, preto boli objekty modelované jednoducho a bez veľkých detailov. Tento modelovací jazyk neposkytuje dostatočné množstvo interakcie, preto by nebol vhodný pre naše virtuálne prostredie. Tieto práce nám však potvrdzujú, že je možné vytvoriť virtuálne prostredie bežiacie vo webovom prehliadači bez herných vývojových prostredí.

3D prostredie nie je potrebné aby hra, zameraná na skladanie počítača, bola užitočná a náučná. To potvrdzuje aj práca [20], v ktorej vytvorili 2D adventúrnu hru. Cieľom hry bolo pozbierať hardvér a poskladať počítač, aby sa mohla deaktivovať bomba a hráč mohol utiecť z miestnosti. Testovali 67 večerných študentov súkromnej univerzity v Taiwane. Z testov pred hrou sa zistilo, že 20 študentov malo skúsenosti so skladaním počítača a 47 študentov nemalo žiadne skúsenosti. Kvôli nečakaným problémom sa záznam z hrania hry uložil iba pre 49 študentov, pričom iba 5 študentov dokončilo hru úspešne. Aj napriek neúspechu veľká časť študentov považovala hru za užitočnú a náučnú, dokonca 85% študentov verilo, že touto hrou sa môžu naučiť viac než z učebníc. Z testov po hraní hry sa zistilo, že študenti s predošlými skúsenosťami sa dokázali z hry naučiť viac než študenti bez skúseností. Ak by mali študenti, ktorý boli neúspešní, možnosť hrať hru znova, tak by sa ich naučené znalosti mohli zvýšiť. Preto v našom virtuálnom prostredí chceme umožniť študentom opakovať úlohy aj po neúspechu, aby sa mohli naučiť viac.

V práci [21] sa použil *depth-first search* algoritmus na výučbu skladania počítača. Vytvorila sa pasívna 3D simulácia, ktorá zobrazovala počítačové komponenty a ich vzájomné pripojenie pomocou animácií. Postupnosť animácií bola vytvorená ako stromová štruktúra, cez ktorú algoritmus prechádzal. Aj napriek zaujímavému prevedeniu, tento spôsob nie je vhodný pre našu prácu, lebo nepodporuje voľnú interakciu s objektmi.

3.4.4 Spojenie videohier a výučby

V práci [9] autori skúmajú problémy a výzvy pri využívaní videohier v školskom vzdelávacom prostredí. Výskum zamerali na výučbu o témach svetové kontinenty a krajiny, prvá pomoc, hardvér a periférie počítača. Študovaná skupina pozostávala zo študentov základných, stredných a vysokých škôl. Hlavné problémy, na ktoré autori upozorňujú, sú dizajn herného prostredia, školská infraštruktúra, povaha učenia, úloha učiteľa, kultúra triedy, angažovanosť študentov. V ďalších odsekoch si ich podrobnejšie rozoberieme.

Vytvorenie každého herného sveta si vyžadovalo približne 60 hodín práce, keďže išlo o kombináciu 3D modelovania, integrácie obsahu a vytvorenia vhodných aktivít. Ďalším problémom bolo vytvorenie vhodného a uveriteľného príbehu pre prostredie a aj napriek veľkej snahe sa študentom podarilo nájsť logické nezrovnalosti v príbehu. Študenti mali na hru vysoké očakávania, porovnávali hru s hrami, ktoré hrajú vo voľnom čase a boli sklamaní, keď niektoré funkcie chýbali. Komplikácie robilo aj ovládanie hry, keďže niektorí študenti nemali skúsenosti s hraním hier a nestačili im inštrukcie v hre. Dizajn hry musel zohľadniť časové obmedzenia výučby čo veľmi ovplyvnilo aktivity, keďže sa všetko muselo stihnúť v určenom čase.

Používanie videohier v školách vyžadovalo prístup k počítačom a internetu. Žiaci nemali vlastné notebooky a preto sa výučba musela presúvať do počítačových učební. Toto však nebolo vždy možné kvôli obsadenosti učební inými triedami. Museli sa preto hľadať alternatívne termíny, čo komplikovalo koordináciu medzi laboratóriom, vyučujúcim a študentmi. Hry sa museli vopred nainštalovať na počítače, aby sa predišlo strate času. Neočakávané problémy ako výpadky internetu alebo zlyhanie počítačov donútili učiteľov improvizovať, aby výučba mohla pokračovať.

Študenti postupovali v hre vlastným tempom. Učiteľ už nepotreboval učiť, ale musel zodpovedať otázky o hre a pomáhať študentom s technickými problémami. Táto „chaotická“ kultúra triedy, kde sa všetci študenti aktívne zapájajú do úloh, spolupracujú a diskutujú o nich, bola učiteľmi akceptovaná a podporovaná, pričom sa snažili udržať rovnováhu medzi voľnosťou a poriadkom.

Hraním hry sa študenti výrazne viac zapojili do výučby a užívali si učenie sa. Dizajn hry však musel udržať v rovnováhe zábavu a vzdelávanie, čo bolo pre niektorých študentov výzvou. Ak sa študenti príliš zameriavali na jednu konkrétnu aktivitu v hre, tak učelia museli zasiahnuť a upriamiť pozornosť na ostatné aktivity. Ak by tak neurobili, tak by študenti nestihli dokončiť výučbu v obmedzenom čase vyhradenom na hodinu.

3.5 Záver systematického prieskumu

Tento systematický prehľad literatúry skúmal využitie virtuálnych prostredí na výučbu o skladaní počítača. Položením našej otázky a s dopytmi z tabuľky 3.2 sme získali 36 unikátnych výsledkov z použitých databáz. Naše výsledky obsahovali aj viaceré prehľady konferencií [8], ktoré neobsahovali žiadne konkrétne informácie, no mali v abstrakte uvedené naše hľadané kľúčové slová. Tieto prehľady sme odignorovali a zhrnuli sme iba práce o virtuálnych prostrediach. Pozrime sa teda na zistenie z našich výsledkov.

Existujú virtuálne prostredia zamerané na výučbu o hardvéri počítača?

Po preštudovaní získaných prác môžeme konštantovať, že existujú virtuálne prostredia zamerané na výučbu o skladaní počítača. Našli sme práce s virtuálnymi prostrediami pre virtuálnu realitu, obohatenú realitu, zmiešanú realitu a aj počítače s klávesnicou a myšou. Tieto prostredia sa preukázali ako efektívny spôsob pre výučbu, s ktorým sú študenti spokojní.

Veľké množstvo prác využívalo herné vývojové prostredia a voľne dostupné 3D modely. Práce, ktoré tieto herné vývojové prostredia nepoužívali, napríklad [18] a [19], nám potvrdili, že vytvoriť virtuálne prostredie je možné bez nich. Avšak kvalita modelov a spôsob interakcií môže byť obmedzený, preto vývoj prostredia bez nich bude náročnejší, s čím musíme počítať.

V jednej práci, [10], bola použitá školská učebňa ako prostredie pre skladanie počítača. Vytvoriť podobné prostredie školskej učebne by mohlo byť užitočné pre ľudí, ktorí by chceli vytvárať prostredia pre náš projekt pre iné školské predmety. Preto zhodnocujeme či by bolo lepšie vytvoriť takéto univerzálne prostredie alebo viac špecializované prostredie pre skladanie počítača, napríklad počítačové servisné centrum.

Prostredia z prác [10], [12] a [13] obsahovali virtuálne postavy, ktoré pomáhali používateľom sa zorientovať v prostredí a poskytovali inštrukcie. Vytvorenie interakcií s takouto postavou môže byť náročné, preto sa musíme rozhodnúť, či chceme takúto postavu v našom prostredí.

Fotorealistické prostredia neboli vždy nevyhnutné. Viaceré menej realistické prostredia, napríklad z prác [16] a [17], mali podobné výsledky z dotazníkov spokojnosti so zážitkom, ako tie fotorealistické, napríklad práca [10]. Toto je dôležitý poznatok, pretože vytvorenie fotorealistického prostredia je náročnejšie

a môže byť nehrateľné na slabších počítačoch. Aj keď fotorealisticnosť prostredia neovplyvnila spokojnosť účastníkov, tak mohla ovplyvniť pocit imerzie a pohltenia prostredím. Preto si myslíme, že fotorealistické prostredie môže byť výhodou, no nie je nevyhnutné.

Našli sme aj prácu [9], ktorá sa zaoberala problémami vzdelávania pomocou videohier. Autori tejto práce zistili, že dizajn 3D prostredia je náročný. Treba udržať rovnováhu medzi zábavou a vzdelávaním a myslieť na časové obmedzenia výučby. Študenti môžu byť sklamaní, ak budú porovnávať naše prostredie s hrami, ktoré boli vyvinuté veľkými hernými štúdiami. Počas výučby formou videohier môžu nastať organizačné a aj technické problémy, ktoré treba riešiť. Študenti pri hraní hry môžu zabudnúť, že sú súčasťou výučby a môže nastať chaotická atmosféra, ktorú treba udržať pod kontrolou. Ak chceme aby študenti stihli všetko v obmedzenom čase, tak treba kontrolovať ich postup a upriamiť ich pozornosť na dôležité aktivity.

Napriek tomu, že veľa prác používalo herné vývojové prostredia, my sme sa rozhodli pracovať s softvérovým rámcom A-Frame, aby sme zachovali doterajšie funkcionality systému. Kvôli tomuto rozhodnutiu očakávame vyšiu náročnosť na vytvorenie prostredia. Rozhodli sme sa, že budeme vyhľadávať voľne dostupné 3D modely, keďže vo viacerých prácach sa ukázalo, že sú dostatočne kvalitné. Budeme sa snažiť nájsť realisticky vyzerajúce modely, ale nie príliš realistické kvôli obavám o výkon. Používateľom chceme v našom prostredí poskytnúť informácie nie len o ovládaní, ale aj o tom ako majú skladať počítač.

4 Návrh z pohľadu používateľa

Komunikáciou so školou, na ktorej budeme prostredie testovať, sme zvolili zameranie virtuálneho prostredia na skladanie počítača. Hlavnou úlohou v prostredí by bolo poskladanie počítača z jednotlivých komponentov. Túto úlohu sme si zvolili, aby sme poskytli študentom možnosť praktického cvičenia zameraného na skladanie počítača z moderných komponentov, ku ktorým škola nemusí mať prístup alebo nechcú riskovať ich poškodenie.

Po ukončení analýzy systematického prehľadu literatúry z kapitoly 3 sme zistili, že tento cieľ je dosiahnuteľný a realizovaný v niekoľkých prácach s využitím rôznych technológií. Primárnou technológiou, na ktorú zameriavame naše prostredie, je rada virtuálnych prilieb Meta Quest¹. Tieto zariadenia máme dostupné na univerzite a môžu byť využité na testovanie prostredia. Veľkou výhodou je aj to, že tieto prilby nepotrebujú počítač na ich prevádzku. Podobne ako v prostrediach, ktoré už máme vytvorené v našom systéme, zachováme možnosť ovládania prostredia aj pre počítače a mobily. Takisto zachováme možnosť úpravy scenára Petriho siete a možnosť lokalizácie textu v prostredí pomocou jazykových súborov.

Výmena komponentov by mohla byť vedľajšia úloha v prostredí. So školou sme prebrali úlohy ako napríklad výmena RAM modulu, alebo výmena grafickej karty. V prostredí by sme mali poskladaný počítač a náhradné komponenty. V scenári vo forme Petriho siete by sa zvolilo, ktorý komponent treba vymeniť. Text úlohy v prostredí môže byť priamy, napríklad „Vymeň RAM modul“, alebo nepriamo naznačiť problém, napríklad „Počítač nič nezobrazuje na monitore“ by naznačoval problém s grafickou kartou.

4.1 3D modely

Počítač sa skladá z komponentov, menších častí, ktoré spolu tvoria celok. Aby sme mohli skladať počítač v 3D prostredí, potrebujeme 3D modely jednotlivých

¹Meta Quest - Dostupná na: <https://www.meta.com/quest/>

komponentov. Vytváranie 3D modelov je zložitý proces, ktorý vyžaduje veľa času a skúseností. Kvôli nedostatku skúseností s 3D modelovaním sme sa rozhodli využiť 3D modely, ktoré sú voľne dostupné na internete. Hľadáme modely vo formáte GLB, lebo ho podporuje softvérový rámec A-Frame. Existuje veľa stránok, ktoré ponúkajú 3D modely na voľné použitie. Najväčšiu ponuku voľne dostupných 3D modelov počítačových komponentov sme našli na stránke Sketchfab². Komponenty, ktoré hľadáme sú:

- Základná doska,
- Procesor,
- Chladič procesora,
- Pamäť RAM,
- Úložisko (M2 SSD, 2,5" SSD, 3,5" HDD),
- Grafická karta,
- Zdroj,
- Počítačová skriňa.

Premýšľali sme aj nad ďalšími komponentami, ako napríklad ventilátory a rôzne rozširujúce karty, ale rozhodli sme sa, že tieto komponenty nie sú nevyhnutné pre náš scenár.

Bolo dôležité vybrať modely, ktoré sú voľne dostupné na edukačné účely. Našli sme všetky modely komponentov, ktoré sme hľadali pod licenciou CC Attribution³. Zoznam modelov, ktoré sme použili, je zobrazený v tabuľke 4.1.

Modely komponentov, ktoré sme našli, mali realistické textúry. Väčšina modelov mala na výber niekoľko variantov textúr, ktoré sa líšili kvalitou. Pre zachovanie použiteľnosti na slabších zariadeniach sme sa rozhodli pre varianty s najnižšou kvalitou textúr. Ako sme spomenuli v kapitole 3.5 pri závere systematického prieskumu, chceli sme nájsť školskú učebnu ako naše virtuálne prostredie. Žiaľ nepodarilo sa nám nájsť vhodný model učebne, ktorého kvalita by bola na úrovni našich počítačových komponentov. Preto sme si zvolili kancelárske prostredie, ktoré bolo vhodné ku našim komponentom.

²Domovská stránka Sketchfab - Dostupná na: <https://sketchfab.com>

³Licensia CC Attribution - Dostupné na: <https://creativecommons.org/licenses/by/>

Autor	URL
Daniel Cardona	https://skfb.ly/6RDsX
Igor.Jop	https://skfb.ly/pqx8z
MUSHROOM_BUILDS	https://skfb.ly/6ZrH9
SrMonteiro	https://skfb.ly/6UKWu
Vivien Deroche	https://skfb.ly/6SNzv
wanderer	https://skfb.ly/oQV6T
wanderer	https://skfb.ly/oRoEF

Tabuľka 4.1: Tabuľka použitých 3D modelov

4.1.1 Úpravy 3D modelov

Stiahnuté 3D modely neboli hneď pripravené na použitie, preto sme sa rozhodli upraviť ich pomocou programu Blender⁴. Boli vyvorené v rôznych veľkostiach, mali rôzne orientácie a počiatkové body. Počiatkový bod je bod, od ktorého sa odvíja pozícia modelu pri vložení do scény na určité súradnice. Okolo tohto bodu sa model otáča, posúva a mení jeho rozmer.

Presunúť počiatkový bod môžeme posunutím modelu voči stredovému bodu, kde sa pretínajú všetky 3 osi. Na začiatku sme chceli normalizovať všetky modely tak, aby ich počiatkový bod bol v ľavom dolnom rohu modelu. Všetky časti modelu by potom smerovali do kladných súradníc od počiatkového bodu. Orientáciu modelov sme upravili tak, aby boli ako keď sú položené na stole, keďže to je najčastejšia pozícia, v ktorej sa počítačové komponenty v scéne budú nachádzať. Táto normalizácia by nám uľahčila presné ukladanie komponentov do scény.

Normalizácia bola jednoduchá pre modely ako je počítačová skriňa alebo zdroj, pretože majú presný roh, podľa ktorého môžeme nastaviť pozíciu. Avšak nie všetky modely majú takýto roh, ako napríklad základná doska, ktorá má zaoblené rohy. Pre riešenie tohto problému sme našli funkciu od autora Shane Ambler pomenovanú `create_bound_box`⁵. Funkcia vytvorí obdĺžnikovú mriežku, ktorá obopína model a vytvorí roh, ktorý môžeme použiť na normalizáciu modelu.

Žiaľ neskôr sme narazili na problém s knižnicou na fyziku a kolízie, ktorý nám dovoľoval pridať kolízny box rozvynutý iba zo stredového bodu modelu. Kolízie objektov potom nefungovali správne, keďže viditeľný model nesedel s kolíznym boxom. Museli sme preto upraviť počiatkové body modelov tak, aby boli v strede

⁴Blender - Dostupný na: <https://www.blender.org>

⁵Funkcia `create_bound_box` - Dostupná na: https://raw.githubusercontent.com/sambler/myblendercontrib/master/create_bound_box.py

modelu.

Keď sme normalizovali všetky počiatočné body, tak sme sa mohli pustiť do zmeny rozmerov modelov. Najprv sme sa pokúsili odhadnúť veľkosť modelov pomocou virtuálnej prilby. Do scény sme vložili rôzne rozmery modelov a voľným okom a meraním od ruky sme odhadovali ich veľkosť. Tento spôsob sa ukázal ako neefektívny, preto sme hľadali iný spôsob.

Zistili sme, že obdĺžniková mriežka, ktorú sme vytvorili na normalizáciu modelov, môže byť použitá aj na presné meranie veľkosti modelov. Mriežka má svoje rozmery uvedené v metroch, takže môžeme presne zmerať veľkosť modelu. Mohli sme tak vytvoriť mriežku pre aktuálny rozmer modelu a mriežku s cieľovými rozmermi, ktoré sme si dohľadali na internete pre každý komponent. Následne sme vedeli vypočítať hodnotu násobku rozmeru pomocou vzťahu 4.1. Tento vzťah sme aplikovali na všetky 3 rozmery každého modelu a zmenili ich veľkosť podľa vypočítaného násobku.

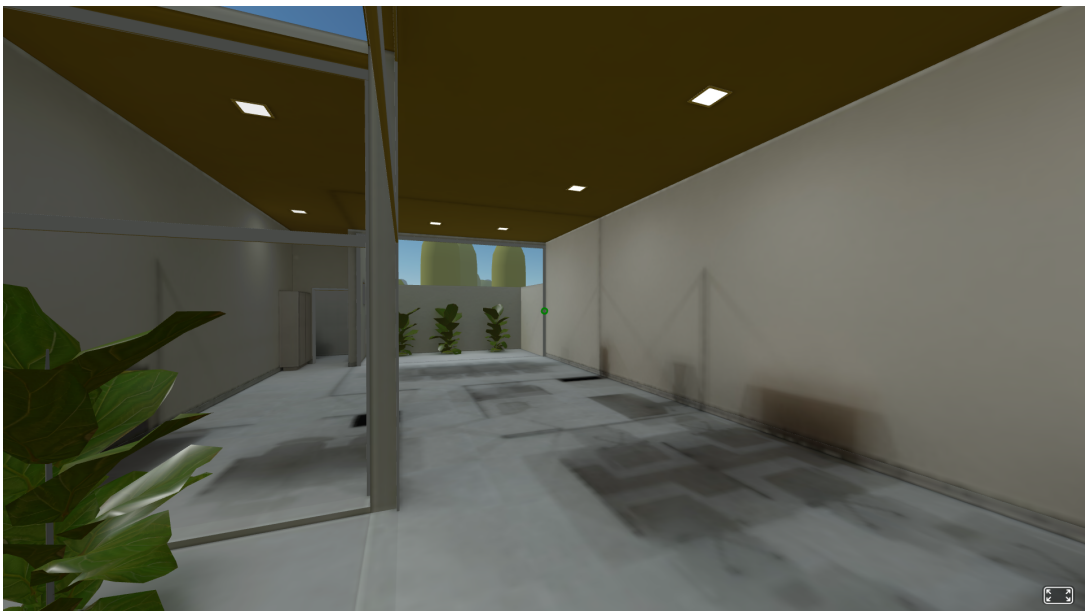
$$\text{násobok rozmeru} = \frac{\text{cieľový rozmer}}{\text{aktuálny rozmer}} \quad (4.1)$$

Zmena veľkosti modelov fungovala, čo bolo jasné viditeľné pri kontrole s VR prílbou. Modely boli v správnych veľkostiach a boli pripravené na použitie v scéne. Dodatočne sme overili kompatibilitu veľkosti modelov pomocou skladania počítača len pomocou úpravy pozície a rotácie modelov. Všimli sme si, že malé komponenty ako je procesor alebo M.2 SSD sú malé pri počítačovej verzii. Ale toto sme nepovažovali za problém, chceli sme zachovať realistické rozmery komponentov.

Model kancelárskeho prostredia, ktorý sme si vybrali nemal realistické rozmery a nevedeli sme si ich ani dohľadať. Preto sme sa rozhodli zmeniť veľkosť modelu približne tak, aby sa hodil k veľkostiam našich modelov počítačových komponentov. Jeho počiatočný bod sme nastavili na pozíciu, na ktorej sme chceli vložiť hráča pri spustení úlohy. Z prostredia sme si oddelili konferenčný stôl, ktorý sme neskôr použili ako pracovné stoly pre jednotlivé úlohy. Pôvodný model si môžete pozrieť na obrázku 4.1. V prostredí bolo veľa objektov, ktoré neboli potrebné pre naše účely. Nechceli sme aby zbytočne zaberali miesto a znižovali výkon, preto sme ich odstránili. Po odstránení nepotrebných objektov nám na textúrach ostali tieňe, ktoré boli vpálené do textúr. Na spomenuté tieňe sa môžete pozrieť na obrázku 4.2. Tieto tieňe sme sa pokúsili odstrániť prekreslením cez textúry. Tieňov sme sa zbavili, ale textúry stále mali vizuálne nedostatky, ktoré sme sa rozhodli ignorovať. Upravený model kancelárskeho prostredia si môžete pozrieť na obrázku 4.3.



Obrázok 4.1: Pôvodný model kancelárskeho prostredia



Obrázok 4.2: Model kancelárskeho prostredia s odstránenými objektami



Obrázok 4.3: Model kancelárskeho prostredia s odstránenými tieňmi

4.2 Scéna

S pripravenými 3D modelmi sme sa mohli pustiť do vytvárania scény. Vložili sme si do scény 3 kancelárske stoly, ktoré sme si upravili na pracovné stoly pre jednotlivé úlohy. Prvý stôl bol určený na skladanie počítača, obsahoval všetky modely komponentov, oddelené od seba. Za stolom je tabuľa, na ktorej sa zobrazuje text úlohy, ktorú má hráč splniť. Tento stôl aj s úlohou je zobrazený na obrázku 4.4.

Ďalšie 2 stoly sú určené na výmenu komponentov. Ľavý stôl obsahuje vopred poskladaný počítač, ktorý je zložený zo všetkých modelov komponentov. Pravý stôl obsahuje náhradné komponenty, ktoré sú určené na výmenu. Tu sme už vynechali modely pre počítačovú skrinku a jej bočné panely, keďže neočakávame, že by ich chceli vymieňať. Na stene za stolmi je tabuľa, na ktorej je zobrazený text úlohy, ktorú má hráč splniť. Tieto stoly aj s úlohou sú zobrazené na obrázku 4.5.

Aby používatelia v prostredí vedeli ako sa pohybovať, presúvať komponenty a celkovo sa orientovať v prostredí, pridali sme uvítaciu tabuľu, ktorá je podobná tej z predchádzajúcej scény z prác Adama Kašelu [1] a Lukáša Pisarčíka [3]. Nájdú tu uvítací text, ktorý im vysvetlí na čo sa toto prostredie zameriava. Nájdú tu aj inštrukcie na ovládanie pomocou klávesnice a myši, alebo pomocou ovládačov virtuálnej reality. Po vstupe do prostredia sa bude používateľ pozeráť priamo na túto tabuľu. Túto tabuľu môžete vidieť na obrázku 4.6.

Ak sa hráč po prečítaní uvítacej tabule otočí, tak uvidí tabuľku uvádzajúcu pôvod 3D modelov. V rámci licencie, pod ktorou sme našli uvedené modely,

sme povinní uviesť autorov modelov. Preto sú uvedený priamo v scéne, ale aj v tabuľke 4.1. Vedľa uvítacej tabule a tabuľky so zásluhami sú vstupné dvere do kancelárie, za ktorými môže hráč vidieť vonkajšie prostredie s trávnikom a stromami. Tabuľku so zásluhami a aj vonkajšie prostredie môžete vidieť na obrázku 4.7.

Pridali sme aj nápomocné plagáty, ktoré môžu pomôcť hráčom bez teoretických vedomostí o počítačoch. Na jednom plagáte máme zobrazené komponenty a ich názvy, aby si hráči vedeli zapamätať názvy komponentov. Komponenty na plagáte sú tie isté ako sa nachádzajú v scéne, aby si ich vedeli zapamätať, ale aj priniesť ku plágotu a identifikovať ich. Na plagát s komponentami sa môžete pozrieť na obrázku 4.8.

Ďalej sme pridali aj plagát, ktorý zobrazuje inštrukcie na spájanie komponentov. Na plagáte je základná doska a počítačová skrinka otočená z oboch strán. Červenou farbou sú označené miesta, kde sa komponenty prichytávajú, pričom od vyznačeného miesta vedie čiara k názvu komponentu. Tento plagát slúži aj ako manuál, ktorý zvykne byť priložený k základnej doske a počítačovej skrinke. Na plagát s inštrukciami sa môžete pozrieť na obrázku 4.9.

Pohľad na hlavnú časť prostredia, kde sa vykonávajú úlohy, je zobrazený na obrázku 4.10. Dodatočne sme si do scény pripravili aj zvuky, ktoré signalizujú prichytenie alebo uvoľnenie komponentu, úspešné splnenie úlohy alebo dokončenie všetkých úloh. Zvuky sme vytvorili sami, nahrávaním zvuku prichytenia a oddelenia magnetu a prichytenia a oddelenia RAM ku základnej doske.

Súbore pre všetky modely a iné assety, ktoré sme použili, nájdete vo výpise 4.1.

Výpis 4.1: Štruktúra súborov použitých assetov v prostredí

```
LirkisEduVePn/LirkisEduVePn-UI/src/assets
├── 3dModels
│   ├── cpu.glb
│   ├── cpu_cooler.glb
│   ├── gpu_amd_rx_6700_xt.glb
│   ├── hdd.glb
│   ├── m2_ssd.glb
│   ├── motherboard.glb
│   ├── office.glb
│   ├── officeNavMesh.glb
│   ├── office_table.glb
│   └── pc_case_left_panel.glb
```

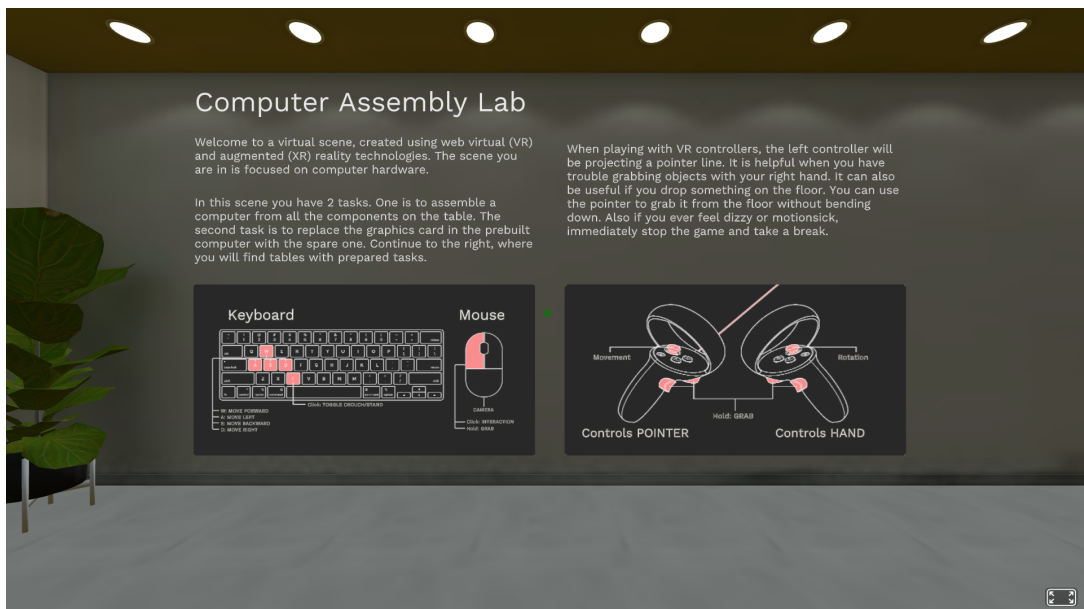


Obrázok 4.4: Úloha na skladanie počítača

```
|   |— pc_case_no_side_panels.glb
|   |— pc_case_right_panel.glb
|   |— psu.glb
|   |— ram.glb
|   |— sata_ssd.glb
|— fonts
|   |— WorkSans-VariableFont_wght-msdf.json
|   |— WorkSans-VariableFontwght.png
|— images
|   |— Keyboard_controls_no_text.png
|   |— Panel.png
|   |— check-circle-fill-green.png
|   |— installation_guide.png
|   |— pc_components_list.png
|   |— vr_controls_no_text.png
|— sounds
|   |— attached.wav
|   |— attached_click.wav
|   |— detached.wav
|   |— detached_click.wav
|   |— success.wav
|   |— win.wav
```



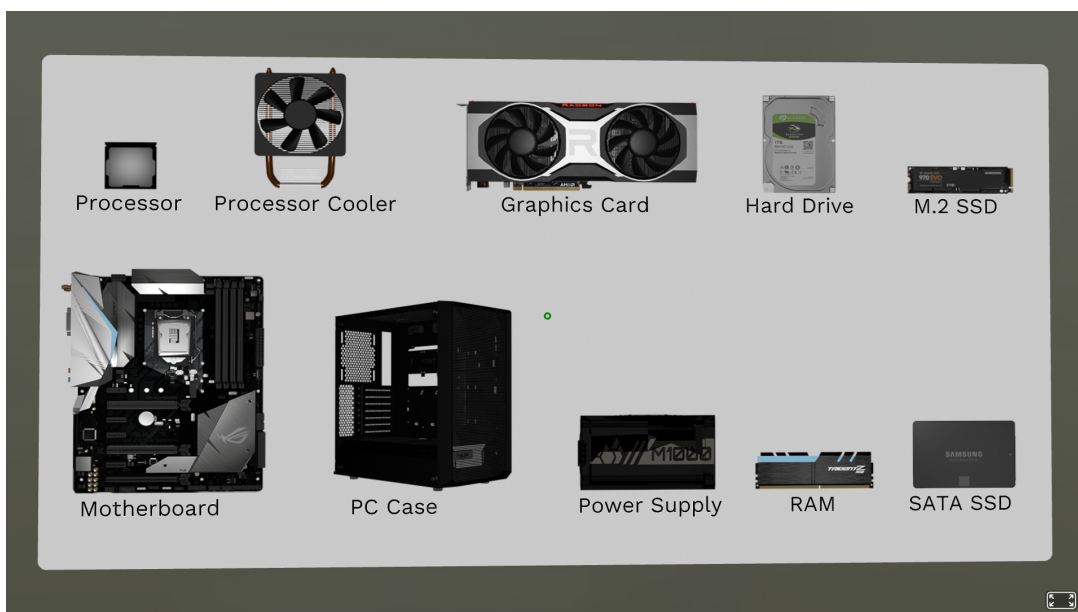
Obrázok 4.5: Úloha na výmenu komponentov



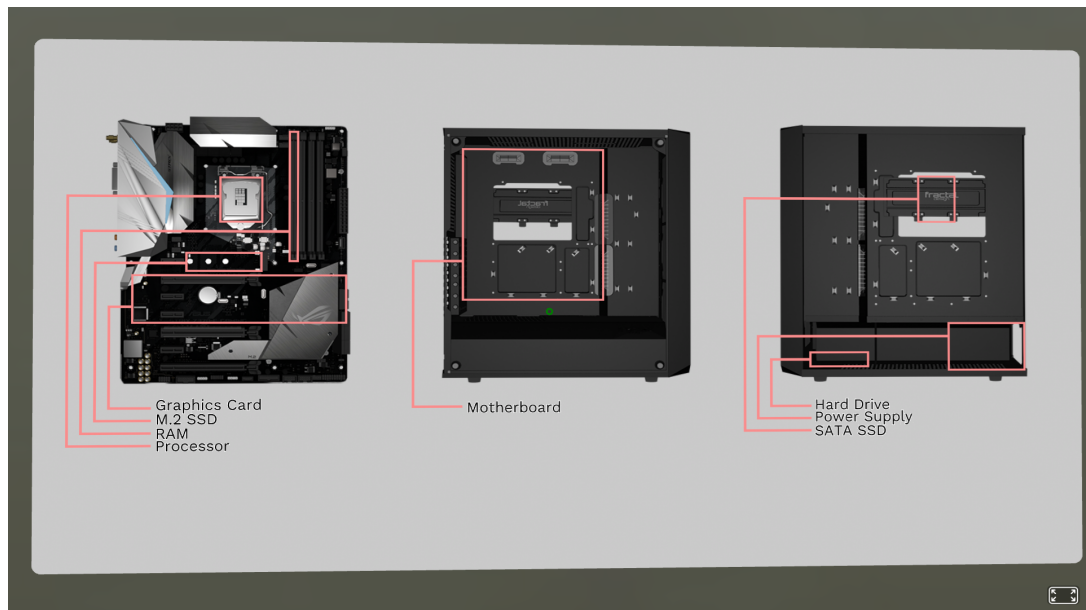
Obrázok 4.6: Uvítacia tabuľa



Obrázok 4.7: Zásluhy za 3D modely



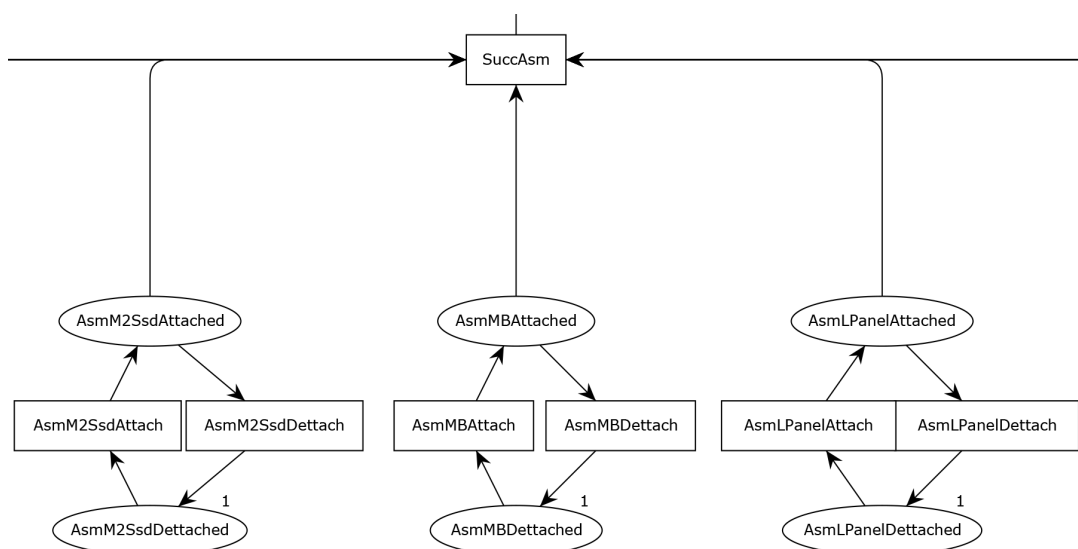
Obrázok 4.8: Plagát s komponentami a ich názvami



Obrázok 4.9: Plagát s inštrukciami na spájanie komponentov



Obrázok 4.10: Model kancelárskeho prostredia vo finálnej scéne



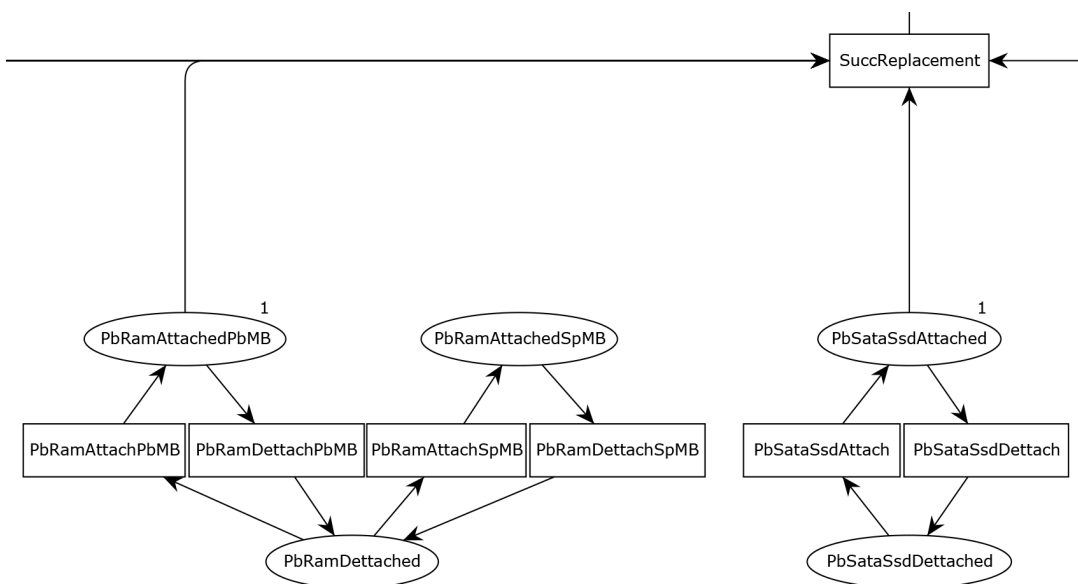
Obrázok 4.11: Časť Petriho siete so scenárom pre skladanie počítača

4.3 Scenáre

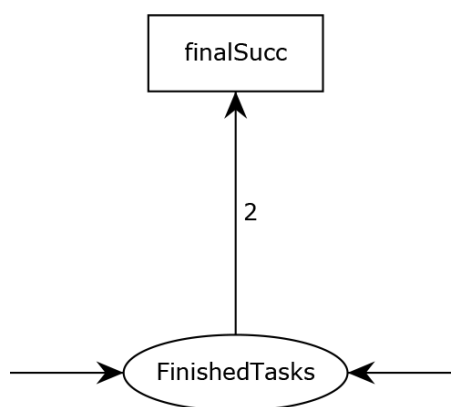
Scenár je tvorený ako Petriho sieť. V scenári sme chceli mať dve odlišné úlohy. Jedna by bola skladanie počítača z komponentov a druhá by bola výmena komponentov vo vopred poskladanom počítači. Vizualnú prípravu pre tieto scenáre môžete vidieť na obrázkoch 4.4 a 4.5.

Vytvorili sme scenár, ktorý bol pomerne jednoduchý a jednoznačne označoval, ktorý komponent je kde prichytený. Scenár je rozdelený na 2 časti, pre úlohu na skladanie počítača a pre úlohu na výmenu komponentov. Prvá časť, označená predponou *Asm*, je určená na skladanie počítača a označuje, ktorý komponent je prichytený a ktorý je voľný. Na obrázku 4.11 je zobrazená časť Petriho siete, ktorá zobrazuje úlohu na skladanie počítača. Druhá časť obsahuje vopred poskladaný počítač, kde sú komponenty označené predponou *Pb*, a náhradné komponenty, ktoré sú označené predponou *Sp*. V druhej časti môže byť väčšina komponentov prichytená buď o komponent z vopred poskladaného počítača alebo o náhradný komponent. Na obrázku 4.12 je zobrazená časť Petriho siete, ktorá zobrazuje úlohu na výmenu komponentov. V celej Petriho sieti, ktorú môžete vidieť v prílohe A, sa opakuje tá istá štruktúra pre rôzne komponenty.

Nastavením šípiek z miest do prechodov *SuccAsm* a *SuccReplacement* si vie tvorca zvoliť aký stav prepojenia komponentov chce, aby hráč dosiahol. Ak niektoré komponenty nie sú potrebné pre vašu úlohu, tak ich časti v Petriho sieti môžete nechať oddelené. Zmenou počiatkových tokenov v miestach si vie nastaviť počiatkový stav prichytenia komponentov. Takýmto spôsobom sa dá vytvo-



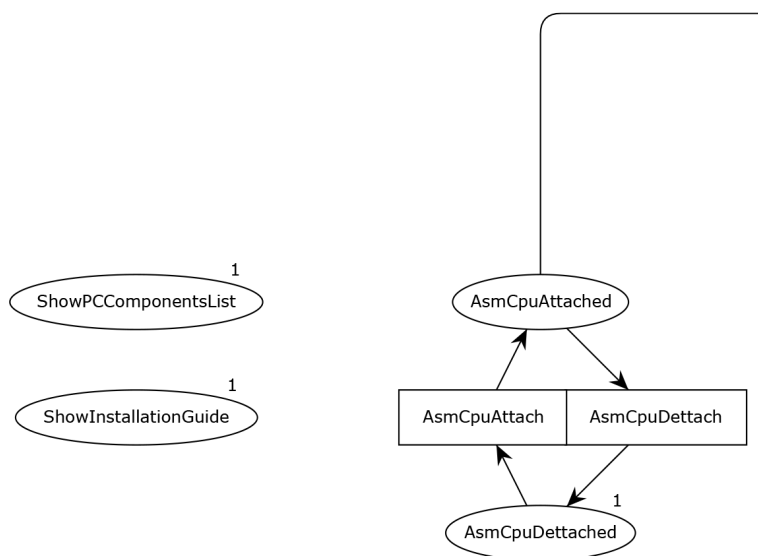
Obrázok 4.12: Časť Petriho siete so scenárom pre výmenu komponentov



Obrázok 4.13: Časť Petriho siete so scenárom pre úspešné splnenie úloh

riť veľké množstvo rôznych úloh, ktoré môžu byť použité na výučbu. Prechody *SuccAsm* a *SuccReplacement* vkladajú tokeny do miesta *FinishedTasks*. Keď sa toto miesto naplní 2 tokenmi, teda hráč splní úlohu na skladanie počítača a aj úlohu na výmenu komponentov, tak sa spustí finálny prechod *finalSucc*, ktorý označuje úspešné splnenie všetkých úloh. Na obrázku 4.13 je zobrazený tento finálny prechod. Tento prechod je zobrazený aj v celej Petriho sieti, ktorú môžete vidieť v prílohe A.

Scenár obsahuje aj 2 izolované miesta, ktoré sú na obrázku 4.14, alebo aj v celej Petriho sieti, ktorú môžete vidieť v prílohe A. Tieto miesta sú určené na zobrazenie nápomocných plagátov. Ich cieľom je dať učiteľom, resp. tvorcom scenárov, možnosť pridať do scény plagáty podľa ich potrieb. Napríklad v prípade, že



Obrázok 4.14: Časť Petriho siete so scenárom pre nápomocné plagáty

sa prostredie použije na hodnotenie vedomostí študentov, tak môžu byť plagáty odstránené, lebo sa očakáva, že študenti už majú potrebné vedomosti. Tieto miesta sa správajú nasledovne:

- *ShowPCComponentsList* - ak má miesto počiatočný token (tzn. token v počiatočnom značení), tak sa v scéne zobrazí plagát s komponentami a ich názvami,
- *ShowInstallationGuide* - ak má miesto počiatočný token (tzn. token v počiatočnom značení), tak sa v scéne zobrazí plagát, ktorý zobrazuje miesta, kde sa majú komponenty prichytiť.

Pre ukážku možnosti úpravy scenárov sme vytvorili 4 rôzne náročné scenáre:

- Jednoduchý - v scéne sa zobrazia oboje nápomocné plagáty, pri skladaní počítača už sú prichytené niektoré komponenty, hráč má vymeniť grafickú kartu, pričom pôvodná grafická karta je už odmontovaná.
- Stredne ťažký - v scéne je iba plagát s prichytením komponentov, žiadne komponenty pre skladanie počítača nie sú prichytené, hráč má vymeniť grafickú kartu, pričom pôvodná grafická karta je stále prichytená.
- Ťažký - v scéne nie sú žiadne plagáty, žiadne komponenty pre skladanie počítača nie sú prichytené, hráč má vymeniť základnú dosku, pričom pôvodná základná doska je stále prichytená, hráč musí presunúť všetky komponenty z pôvodnej základnej dosky na novú základnú dosku.

- Testovací - v scéne sú oboje nápomocné plagáty, žiadne komponenty pre skladanie počítača nie sú prichytené, hráč má vymeniť grafickú kartu, pričom pôvodná grafická karta je stále prichytená, tento scenár sme použili na naše testovanie prostredia v kapitole 6.

Všetky súbory pre tieto scenáre sú v projekte vo svojich vlastných priečiňkoch, ktoré môžete vidieť vo výpise 4.2.

Výpis 4.2: Štruktúra súborov scenárov

```
LirkisEduVePn/LirkisEduVePn-UI/src/app/components/tasks/
scenarios/computer_assembly
├── README.md
├── easy
│   ├── computer_assembly_easy.cpn
│   └── lang
│       ├── sceneDescription.en.json
│       └── sceneDescription.sk.json
├── hard
│   ├── computer_assembly_hard.cpn
│   └── lang
│       ├── sceneDescription.en.json
│       └── sceneDescription.sk.json
├── normal
│   ├── computer_assembly_normal.cpn
│   └── lang
│       ├── sceneDescription.en.json
│       └── sceneDescription.sk.json
└── testing
    ├── computer_assembly_testing.cpn
    └── lang
        ├── sceneDescription.en.json
        └── sceneDescription.sk.json
```

5 Implementácia

V tejto kapitole si rozoberieme implementáciu interakcií v našej scéne. Návrh samotnej scény sme popísali v sekcii 4.2. Scenáre sme navrhli v sekcii 4.3. Na výpise 5.1 vidíme časť štruktúry súborov, ktoré využívame v našej scéne:

Výpis 5.1: Štruktúra súborov použitých na vytvorenie prostredia pre skladanie počítačov

```
LirkisEduVePn/LirkisEduVePn-UI/src/app/components/tasks
├── js
│   ├── components
│   │   ├── attachmentBox.component.js
│   │   ├── blinkControl.component.js
│   │   ├── computerAssemblyCollisionGroup.component.js
│   │   ├── crouch.component.js
│   │   ├── mobileViewHandler.component.js
│   │   ├── petriNetSim.component.js
│   │   ├── phaseShiftHands.component.js
│   │   ├── pn-transition-handler.component.js
│   │   ├── sceneLanguage.component.js
│   │   ├── transitions.ts
│   │   ├── visibilityBasedOnPetriNet.component.js
│   │   └── vrDetectionForRaycasting.component.js
│   └── modules
│       ├── petriNet.mjs
│       ├── petriNetLoader.mjs
│       └── userActivityLogger.js
└── scenes
    └── computer_assembly
        ├── computer_assembly.component.css
        ├── computer_assembly.component.html
        ├── computer_assembly.component.spec.ts
        └── computer_assembly.component.ts
```

```

└─ scripts
    └─ sceneScript.ts

```

Hlavné interakcie, ktoré potrebujeme, sú spájanie a odpájanie objektov. Pre to potrebujeme zistiť ako funguje detekcia kolízií a ako vieme spojiť 2 objekty dokopy. V predošlej scéne bol vyhradený statický box, ktorý kontroloval kolízie objektov, ktoré sa do neho dostali. Spojenie objektov sa dalo riešiť vnorením jedného objektu do druhého, alebo pridaním spoločného objektu, ktorý by ich spájal. Pri vnorených objektoch je ale potrebné riešiť zmenu pozície z absolútnej na relatívnu, ktorá závisí od pozície rodičovského objektu.

Počas diskusie s vedúcim práce a Kristiánom Zsigóm [5], ktorý pracoval na virtuálnom prostredí pre ukážku možných funkcionality pre náš projekt, sme dostali návrh na použitie fyzikálnej knižnice *Ammo.js*⁶. Rozhodli sme sa ho využiť a pretvoriť našu scénu podľa *showcase_room* scény, ktorú vytvoril Kristián Zsigó.

Využitie novej fyzikálnej knižnice nám pridalo realistickú fyziku a kolízie. Naše objekty boli ovplyvnené simulovanou gravitáciou a ich kolízie boli nie len detekované, ale aj simulované, čo pôsobilo veľmi realisticky. Na spájanie objektov sme mohli použiť komponent *ammo-constraint* priamo z tejto knižnice.

5.1 Kolízie objektov

Naším počítačovým komponentom sme potrebovali pridať kolízne boxy. Vo fyzikálnej knižnici je mnoho možností, ktoré by vedeli vytvoriť kolízne boxy automaticky pre naše komplexné modely. Avšak všetky tieto možnosti boli príliš výkonnovo náročné, čo spôsobilo, že naša scéna nebola použiteľná. Preto sme použili obdĺžnikové kolízne boxy, ktoré nemali taký dopad na výkon.

Avšak toto riešenie vytvorilo problém, lebo sme nemohli vkladať komponenty do seba, napríklad kolízny box skrine nás nepustil vložiť matičnú dosku dovnútra. Na riešenie tohto problému sme vytvorili komponent *computerAssemblyCollisionGroup*, ktorý podľa parametra *group* nastavil kolízne skupiny a kolízne masky pre jednotlivé objekty. Pre zachovanie kolízií medzi čo najväčším počtom objektov sme sa rozhodli vytvoriť každému komponentu vlastnú skupinu a masku. Ak sa kolízne boxy pri poskladanom počítači museli prekrývať, tak ich kolízie boli vypnuté, inak by sa objekty od seba odpudzovali. Okrem skupín pre komponenty sme pridali aj špeciálne skupiny:

- *default* - pre objekty prostredia ako sú steny, stoly, podlaha a strop,

⁶Ammo.js - Dostupné na: <https://n5ro.github.io/aframe-physics-system/AmmoDriver.html>

- *hand* - pre ruky pri použití VR ovládačov,
- *attachmentBox* - pre kolízne boxy používané na spájanie objektov, napr. vloženie matičnej dosky do skrine.

Komponent tiež zabezpečuje aby sa kolízne boxy GLTF modelov pridali až po načítaní modelov. Pridanie kolíznych boxov pred načítaním modelov by spôsobilo, že by nemali správne rozmery kvôli čomu sa neposielali správy o kolíziách. Vopred definované skupiny majú vybrané statické alebo aj kinematické kolízne telá, ostatným objektom sa nastaví dynamické kolízne telo. Pre objekty, ktoré majú v sebe vnorené objekty, sa môže nastaviť kolízne telo manuálne pomocou parametra *manualFit* a využitia buď parametra *halfExtents* na obdĺžnikové kolízne telo, alebo parametrov *heightfieldDistance* a *heightfieldData* na kolízne telo typu *heightfield*. Manuálne nastavenie kolízneho boxu sme využili napr. na počítačovú skriňu a matičnú dosku. Pridali sme aj parameter *initializeAfter*, ktorým zabezpečujeme, že sa dynamické kolízne telo priradí až po prijatí správy o úspešnej inicializácii vybraného elementu. Zabezpečuje to napríklad, že stôl sa načíta skôr ako objekty na ňom, inak by sa objekty prepadli cez stôl.

Problém s kolíznymi boxami bol aj pre kancelárske prostredie. Tu sme už mohli použiť komplexnejšie kolízne boxy, ktoré sa automaticky mapovali na tvar modelu. Keďže prostredie bolo statické a nemuselo sa pohybovať, tak to nemalo taký dopad na výkon. Avšak automaticky generované kolízne boxy neboli spoľahlivé a naše komponenty sa mohli zaseknúť v stenách alebo prepadnúť cez podlahu. Kolízne boxy boli príliš tenké, preto keď pohybujúci sa objekt nabral rýchlosť, tak sa vedel čiastočne cez neho prebiť, čo spôsobovalo nepredvídateľné správanie, ktoré nakoniec skončila stratou objektu mimo scénu. Vyriešili sme to pridaním neviditeľných boxov s kolíznymi boxami, ktoré sa položili na miesta, kde majú byť steny, podlaha a strop. Dbali sme na to, aby mali boxy dostatočnú hrúbku, aby sa objekty cez nich nemohli prebiť.

5.2 Spájanie objektov

Spájanie objektov sme chceli riešiť pomocou A-Frame komponentu *ammo-constraint* z fyzikálnej knižnice *Ammo.js*. Vytvorili sme si vlastný komponent *attachment-box*, ktorý sme pridávali na kolízne boxy, ktoré boli určené na spájanie objektov. Boxu sme nastavili rozmery podľa veľkosti objektu, ktorý mal byť do neho vložený. Komponentu sme potom poskytli parametre o aký objekt sa má prichytiť, relatívnu pozíciu a rotáciu od daného objektu, skupinu objektov, ktoré sa mô-

žu prichytiť a zoznam povolených identifikátorov objektov.

Náš komponent nastavil správnu pozíciu a rotáciu vzhľadom na objekt, o ktorý sa mal prichytiť. Pre box sa potom nastavilo dynamické kolízne telo so skupinou *attachmentBox* a maskou, ktorá obsahovala iba skupinu povolených objektov na prichytenie. Potom sme mohli prichytiť box o objekt pomocou *ammo-constraint*. Ak nastala kolízia s objektom so správnou skupinou a identifikátorom, tak sa vypla detekcia kolízií pre box, objektu sa nastavila pozícia a rotácia podľa boxu a prichytil sa o objekt tak isto ako bol prichytený box.

Ak používateľ držal objekt v momente prichytenia objektu, tak sa spojené objekty začali nekontrolovateľne pohybovať a otáčať. Tomuto problému sme sa dokázali vyhnúť ak bol objekt uvoľnený nad miestom prichytenia, tak jeho pád ho posunul na správne miesto a prichytenie prebehlo bez problémov.

Prichytené objekty neboli stabilne prichytené počas pohybu. Napríklad ak používateľ zobral matičnú dosku s prichytenou RAM pamäťou a začal sa s ňou prechádzať po prostredí, tak RAM pamäť počas pohybu vychýlila a nestíhala sa presunúť na správnu pozíciu. Ak sa používateľ zastavil, tak sa po chvíli RAM pamäť dostala na správnu pozíciu.

Komponenty prichytené zvislo do počítačovej skrine, hlavne matičná doska, grafická karta, chladič procesora a SATA SSD, mali vychýlenú rotáciu. Testovaním prichytenia týchto komponentov sme zistili, že nebol problém s nastavenými parametrami prichytenia, ale s vplyvmi fyzikálneho modulu na objekty.

Riešenie pre spomenuté problémy sme nenašli na internete, tak sme sa rozhodli experimentovať s rôznymi nastaveniami fyzikálneho modulu. Začali sme testovaním rôznych typov prichytenia pre *ammo-constraint*. Zistili sme, že predvolená hodnota *lock*, ktorú sme používali, bola najvhodnejšia pre naše účely, pretože ostatné typy prichytenia zhoršovali stabilitu prichytených objektov. Skúsili sme vypnúť vplyv gravitácie na prichytené objekty, ale to nám tiež nepomohlo. Menili sme váhu objektov najprv na realistické hodnoty, neskôr na minimálne hodnoty, ale to nám tiež nepomohlo.

Experimentovali sme s vnorenými objektami. Zistili sme, že ak vytvoríme vnorený objekt, kde rodičovský objekt má dynamické kolízne telo a vnorený objekt má kinematické kolízne telo, tak sú tieto objekty stabilne prichytené a dajú sa posúvať chytením rodičovského objektu. Vytvorili sme si preto kolíznu skupinu *container*, ktorá slúži ako rodičovský objekt pre skupinu objektov, ktoré chceme prichytiť. Prerobili sme počítačové komponenty v našej scéne tak, aby sme mali vnorené objekty. Použili sme 3 vnorené kontajnerové objekty:

- hlavný kontajner pre celý počítač, hlavný objekt je počítačová skrinka a ob-

sahuje komponenty, ktoré sa o ňu prichytia a zároveň kontajner pre maticnú dosku,

- kontajner pre maticnú dosku, jej komponenty a kontajner pre procesor,
- kontajner pre procesor a jeho chladič.

Náš komponent *attachment-box* sme prerobili tak, aby namiesto *ammo-constraint* prichytil objekt len zmenou z dynamického na kinematické kolízne telo. Pozície a rotácie sa presunuli z komponentu priamo do definície objektu, vďaka čomu sme zredukovali počet parametrov komponentu.

Táto zmena vyriešila spomínané problémy. Kinematické objekty sa nedajú presúvať používateľom priamo, preto keď sa objekt pripojí tak sa rovno pustí, čo vyriešilo problém pri pripájaní objektov. Spojené objekty sú veľmi stabilné, ani pri pohybe nie je vidieť žiadne vychýlenie. Ale pri testovaní sme narazili na problém, že ak je rodičovský objekt otočený, tak pri presúvaní sa vnorené objekty zvláštne otáčajú. Otáča sa iba ich 3D model a to okolo pozície rodičovského objektu, pričom kolízny box vnoreného objektu sa otáča podľa očakávaní. Takto vzniká veľká odchýlka medzi kolíznym boxom a 3D modelom, kvôli ktorej objekty môžu vyzeráť a správať sa zvláštne ak sú napríklad položené na stole. Tento problém by bol pre používateľov veľmi mätúci, tak sme hľadali iné riešenie.

Spomenutý problém sme identifikovali a zistili sme, že vznikol kvôli neustálenému vnoreniu objektov do kontajnerov. Rotácia kontajnera ovplyvňuje odchýlku rotácie vnoreného objektu. Riešením bolo dynamické vnáranie objektov, preto sme potrebovali zistiť ako zmeniť rodičovský objekt vnoreného objektu.

Zmenu rodičovského objektu sme skúšali riešiť presunom HTML elementu do iného elementu. Tento prístup nefungoval, lebo komponenty pripojené ku presunutému objektu boli zmätené a prestali fungovať. Zistili sme, že náš softvérový rámec *A-Frame* používa na pozadí knižnicu *three.js*⁷. Vyskúšali sme zmeniť rodičovský objekt pomocou funkcie *attach* z knižnice *three.js*, čo fungovalo. Vyriešilo to náš problém s rotáciou vnorených objektov. Po spomenutej úprave sme už nepotrebovali skupinu *container*, preto sme ju odstránili spolu s parametrami, ktoré využívala.

5.3 Odpájanie objektov

V našom komponente *attachment-box* sme pridali možnosť odpájania objektov. Po pripojení objektu začneme poučúvať na správy o chytení objektu. Ak používateľ

⁷Knižnica *three.js* - Dostupná na: <https://threejs.org>

pripojený objekt chytí, nastavíme mu dynamické kolízne telo vďaka čomu sa objektom môže manipulovať. Objektu zmeníme rodičovský objekt na hlavný objekt scény, aby neostal vnorený, čo by spôsobovalo problém s rotáciou. Keď používateľ objekt pustí, tak povolíme opätovné prichytenie objektu a prestaneme počúvať na správy o chytení objektu.

Pri odpájaní objektov sme narazili na problém, že sa objekt hneď po odpojení prepadol cez podlahu. Identifikovali sme, že tento problém bol spojený s použitím funkcie *attach* na prichytenie objektu a následnou zmenou kolízneho tela na dynamické. Tento problém bol spomenutý na GitHub repozitári fyzikálnej knižnice v issue #131⁸, kde identifikoval problém s funkciou *attach* a našli riešenie vnorením objektu do prázdneho elementu, a v issue #138⁹, kde identifikovali problém s prepadávaním objektu cez podlahu, ale nepopísali jeho riešenie. Vyskúšali sme pridať prázdny element do scény a vnoriť do neho objekt, ktorý sa mal prichytiť a toto riešenie fungovalo.

Odpájanie objektov fungovalo dobre, kým neboli vložené do počítačovej skrine. Keď sa objekt nachádzal vnútri skrine, tak sa ťažko chytil. Vyriešili sme to zmenou tvaru kolízneho boxu skrine z obdĺžnikového na tvar typu *heightfield*. Tento tvar sa vytvára mriežku bodov, ktorým sa nastaví výška. My sme tento tvar nastavili tak, aby kolízny box zachytil iba prednú a zadnú časť skrine.

Po spomenutých úpravách sa už dali všetky objekty odpájať. V niektorých prípadoch nastal problém, že sa chytil nesprávny objekt, ale pri pokuse o chytenie objektu z iného uhla sa už chytil správny objekt. Tento problém nenastal často, tak sme sa rozhodli ho neriešiť.

5.4 Scenáre vo forme Petriho sieti

Návrh scenárov sme popísali v sekcii 4.3. Na prepojenie scény so scenárom Petriho sieti sme využili komponent *pn-transition-handler* z práce Kristiána Zsiga [5]. Tiež sme využili jeho dve základné rozhrania *BaseTransition* a *BasePlace*, podľa ktorých sme vytvorili triedy *UniversalTransition*, *FinishTransition*, *UniversalPlace*. Trieda *FinishTransition* je jedinečná tým, že obsahuje identifikátory pre indikátor dokončenia a indikátor nedokončenia úlohy. Keď sa prechod odpáli, tak sa zobrazí indikátor dokončenia úlohy, pošle sa správa na prehratie zvuku a indikátor nedokončenia úlohy sa skryje.

⁸GitHub issue #131 - Dostupné na: <https://github.com/n5ro/aframe-physics-system/issues/131>

⁹GitHub issue #138 - Dostupné na: <https://github.com/n5ro/aframe-physics-system/issues/138>

Komponent *petriNetSim*, ktorý je určený na prácu so scenármi Petriho sieťami, sme rozšírili o správu, ktorá sa pošle pri načítaní scenára, alebo v prípade opätovného spustenia po odpálení prechodov z predchádzajúceho sedenia. Táto správa obsahuje aktuálny stav Petriho siete. Túto správu využívame v komponente *attachmentBox* na inicializáciu objektov, ktoré sa majú prichytiť. Podľa parametrov *petriNetPlaces* a *preattached* si vyhľadá miesta v Petriho sieti, a ak majú v sebe tokeny, tak vyberie objekty, ktoré sa majú prichytiť. Vďaka tomu vieme upravovať počiatočný stav komponentov v scéne bez toho, aby sme museli priamo zasahovať do scény. Spomenutú správu sme využili aj na zobrazovanie, resp. skrytie, objektov v scéne. Na objekt v scéne sa pridá komponent *visibilityBasedOnPetriNet* s parametrom *place*, a po prijatí správy sa v Petriho sieti vyhľadá miesto s daným identifikátorom. Ak sa miesto v sieti nenachádza, alebo nemá tokeny, tak sa objekt skryje.

5.5 Úpravy pre zlepšenie použiteľnosti

Počas skúšania našej scény sme si všimli nedostatky s výškou hráča. Pri ovládaní s VR prilbou bol používateľ príliš vysoký a musel sa zohýbať ku stolom. Po úprave jeho výšky bolo ovládanie s VR prilbou pohodlnejšie. Ale pri ovládaní cez počítač bol hráč stále príliš vysoký a nemal dobrý pohľad na všetky komponenty. Pridali sme preto komponent *crouch*, ktorým vieme zmeniť výšku hráča stlačením zvolenej klávesy.

Používateľ sa už vedel dostatočne priblížiť ku komponentom, ale niekedy mal problém s nedostatkom svetla v scéne. Skúsili sme pridať svetlo ku stolom v scéne, ale vo VR približne *Meta Quest 2* to viditeľne ovplyvnilo výkon. Keďže tento problém ovplyňoval iba zopár počítačových komponentov a svetlo výrazne ovplyvnilo výkon, rozhodli sme sa svetlá odstrániť a tento problém neriešiť. Možným riešením by bolo pridať čelovú lampu, ktorú by si používateľ mohol zapnúť alebo vypnúť. Keďže by nebola neustále zapnutá, tak by možno nemala taký dopad na výkon.

S VR prilbou vie používateľ voľne manipulovať s objektmi v scéne. Vie si ich prisunúť bližšie, alebo ich odsunúť ďalej, otáčať ich a nakláňať. Všetky tieto akcie sú veľmi intuitívne a prirodzené. Ale pri ovládaní cez počítač je manipulovanie s objektmi veľmi obmedzené. Hráč ich vie iba chytiť a posunúť na iné miesto. Objekty si držia rovnakú vzdialenosť a orientáciu, akú mali pred chytením. Toto správanie je veľmi neprirodzené a obmedzujúce. Chceli sme preto pridať možnosť prisunúť alebo odsunúť objekt od hráča, keď je chytený. Ale nepodarilo sa nám

nájsť spôsob, ako to spraviť, lebo chytené objekty sa nehceli posúvať. Veríme, že tento problém by sa dal vyriešiť, ale nezostal nám čas na jeho riešenie. Takisto sme už nemali čas na riešenie problému s otáčaním objektov pri ovládaní cez počítač.

6 Vyhodnotenie

V tejto kapitole popisujeme priebeh testovania virtuálneho prostredia na žiakoch strednej školy. Preberieme si dotazníky, ktoré sme použili na vyhodnotenie testovania. Na záver si prejdeme vyhodnotením výsledkov z dotazníkov.

Pred testovaním najprv subjektívne zhodnotíme naše očakávania od testovania. Očakávame, že testovanie s VR prilbou bude pre žiakov zábavnejšie a pútavejšie ako klasické testovanie na počítači. Očakávame, že účastníci testovania, ktorý ešte nemali skúsenosti so skladaním a rozoberaním počítačov, budú mať zvýšenú istotu pri skladaní počítača. Predpokladáme mierne sklamanie z nedostatku interakcie s objektmi pri používaní počítačovej verzie, z kvality vizuálnej stránky prostredia a počítačových komponentov, a z neúplnosti skladania počítača, konkrétne z chýbania možnosti prepájať káble alebo zapnúť počítač pre skúšku funkčnosti. Predpokladáme že VR prilby budú problematické, keďže očakávame, že s nimi žiaci nebudú mať skúsenosti. Môžu mať problémy s nastavením rozsahu šošoviek pre ich zrak, prípadne problémy s ovládaním pomocou ovládačov. Predpokladáme že približne polovica žiakov bude mať problémy s rozmazaným videním vo VR prilbe. Očakávame, že malé percento žiakov môže pocítiť nevoľnosť alebo závraty.

6.1 Príprava testovania

Pre testovanie sme vytvorili vlastný scenár, označený ako testovací scenár. V scéne sú 2 nápomocné plagáty. Jeden plagát obsahuje komponenty počítača a ich názvy. Druhý plagát obsahuje inštrukcie na skladanie počítača, konkrétne kde sa majú komponenty prichytiť. Pre úlohu skladania počítača je potrebné poskladať počítač zo všetkých komponentov na stole, pričom žiadne komponenty nie sú vopred prichytené. Druhou úlohou je vymeniť grafickú kartu, pričom na stole je vopred poskladaný počítač. To znamená, že účastníci testovania musia najprv odobrať bočný panel počítača, následne odobrať pôvodnú grafickú kartu, prichytiť novú grafickú kartu a nakoniec prichytiť bočný panel počítača naspäť. Viac

o scenároch sme popísali v sekcii 4.3.

Účastníkom testovania sme pripravili používateľské účty do systému, aby sa nemuseli registrovať. Chceli sme čas testovania skrátiť na minimum.

So školou, Súkromná stredná odborná škola ekonomická KOŠICKÁ AKADÉMIA, sme sa dohodli na testovaní 29. 4. 2025. Škola nám poskytla počítačovú miestnosť na 2 vyučovacie hodiny (1 vyučovacia hodina = 45 minút). Testovania sa zúčastnila trieda 3. ročníka a trieda 4. ročníka z odboru informačné systémy a služby. Počas 20 minútovej prestávky medzi vyučovacími hodinami sme poskytli možnosť vyskúšať prostredie aj pre 2 žiakov, ktorí prejavili záujem. Dodatočne sme testovali rodinu a priateľov, ktorí mali záujem o testovanie.

Aby sme mohli testovanie vyhodnotiť, potrebujeme získať spätnú väzbu od účastníkov testovania. Preto sme pre nich pripravili anonymný dotazník, ktorý vyplnili po testovaní. Dotazník bol poskytnutý v slovenskom jazyku. Niektorí účastníci testovania pochádzali z iných krajín, ale študujú na škole v slovenčine, preto sme sa rozhodli dotazník poskytnúť v iba slovenskom jazyku. Dotazník mal nasledujúcu štruktúru:

- demografické otázky o účastníkovi,
- 2 voľné otázky na hodnotenie prostredia,
- *System Usability Scale* (SUS) [22] - na hodnotenie vnímanej použiteľnosti prostredia,
- *NASA Task Load Index* (NASA TLX) [23] - na hodnotenie pracovnej záťaže,
- *Igroup Presence Questionnaire* (IPQ) [24] - na hodnotenie prítomnosti vo virtuálnom prostredí,
- *Virtual Reality Sickness Questionnaire* (VRSQ) [25] - na hodnotenie nevoľnosti vo virtuálnom prostredí,

Vedúci práce, Ing. Štefan Korečko, PhD., nám poskytol preklady všetkých 4 dotazníkov do slovenského jazyka. Otázky z dotazníkov IPQ a VRSQ boli iba pre skupinu účastníkov, ktorý testovali prostredie s VR prilbou. Otázky v dotazníku mali nasledujúce znenie:

Demografické otázky

1. Identifikátor z prihlasovacích údajov (napr. ak ste dostali email p01@gmail.com, tak zadajte 01)

2. Pohlavie
3. Vek
4. Zameranie
5. Mali ste predchádzajúcu skúsenosť s virtuálnou realitou?
6. Mali ste predchádzajúcu skúsenosť so skladaním počítača?
7. Ako ste si vyskúšali scénu?

Voľné otázky

1. Vadilo ti niečo na systéme?
2. Páčilo sa ti niečo na systéme?

System Usability Scale (SUS)

1. Myslím si, že by som rád(a) používal(a) tento systém často.
2. Systém mi prišiel zbytočne komplikovaný.
3. Systém sa mi zdal ľahko použiteľný.
4. Myslím si, že by som potreboval(a) technickú podporu aby som vedel(a) používať tento systém.
5. Funkcionality systému sú podľa mňa dobre zapracované (integrované).
6. Zdalo sa mi, že v tomto systéme je priveľa nekonzistentností (nezrovnalostí).
7. Viem si predstaviť, že väčšina ľudí by sa tento systém naučila veľmi rýchlo používať.
8. Tento systém mi príde ťažkopádny na použitie.
9. Pri používaní systému som sa cítil(a) sebaisto.
10. Než som mohol(la) začať pracovať s týmto systémom, musel(a) som sa naučiť veľa vecí.

NASA Task Load Index (NASA TLX)

1. Mentálne zaťaženie: Ako mentálne náročné bolo plnenie úloh? (0 - 100)
2. Fyzické zaťaženie: Ako fyzicky náročné bolo plnenie úloh? (0 - 100)
3. Časový nátlak: Ako uponáhľané (pod tlakom) bolo plnenie úloh? (0 - 100)
4. Výkon: Ako úspešní ste boli pri plnení úloh? (0 - 100)
5. Úsilie: Ako ťažko ste museli pracovať aby ste dosiahli to, čo sa vám podarilo? (0 - 100)
6. Frustrácia: Ako veľmi ste boli neistí, znechutení, podráždení, vystresovaní a otrávení? (0 - 100)

Igroup Presence Questionnaire (IPQ)

1. V počítačom generovanom svete som mal(a) pocit, ako keby som "bol(a) tam".
2. Mal(a) som pocit, že ma virtuálny svet obklopuje.
3. Mal(a) som pocit, že vnímam (vidím) iba obrázky.
4. Necítil(a) som sa vo virtuálnom svete prítomný(á).
5. Mal(a) som pocit, že naozaj konám vo virtuálnom svete a nie, že ho zvonku ovládam.
6. Cítil(a) som sa vo virtuálnom svete prítomný(á).
7. Do akej miery ste si boli vedomí reálneho sveta (teplota, zvuky, iní ľudia,...) počas vašej aktivity vo virtuálnom svete?
8. Neuvedomoval(a) som si okolitý reálny svet.
9. Stále som venoval pozornosť reálnemu prostrediu.
10. Virtuálny svet úplne upútal moju pozornosť.
11. Ako veľmi reálny vám pripadal virtuálny svet?
12. Ako veľmi sa zážitok vo virtuálnom svete zhodoval so zážitkom v reálnom svete?

13. Nakoľko reálny sa vám zdal virtuálny svet?
14. Virtualny svet sa mi zdal skutočnejší ako ten reálny.

Virtual Reality Sickness Questionnaire (VRSQ)

1. Všeobecný diskomfort
2. Únava
3. Námaha očí
4. Ťažkosti so zaostrením
5. Bolesť hlavy
6. Tlak v hlave
7. Rozmazané videnie
8. Závraty pri zatvorených očiach
9. Vertigo (točenie hlavy)

Na testovanie sme sa rozhodli použiť lokálne bežiaci server kvôli zabezpečeniu úplnej kontroly a dostatočného výkonu.

6.2 Priebeh testovania

Na testovanie sme prišli 5, aby sme mali dostatok ľudí na pomoc s VR prilbami a inými problémami. Pomôcť nám prišiel vedúci práce Ing. Štefan Korečko, PhD., jeho syn, Roland Korečko, moja sestra, Viktória Sakáčová, a Bc. Dávid Teplan. Po príchode do školy sme sa stretli s Mgr. Michaelou Haluškovou, ktorá nám pomohla s organizovaním testovania.

Ako server sme použili náš notebook, ktorý sme pripojili na školskú sieť. Na účely testovania sme priniesli 2 VR prilby Meta Quest 2 a 1 VR prilbu Meta Quest Pro. VR prilby sme pripojili na školskú sieť cez Wi-Fi. Po spustení scény na VR prilbách sme si všimli, že pohyb v scéne veľmi seká. Bolo to spôsobené úvodným načítavaním scény. Tento problém sme si v domácom prostredí nevšimli, keďže sme mali rýchlejšie pripojenie k sieti. Načítavanie scény cez školskú sieť trvalo približne minútu. Po načítaní scény už bol pohyb plynulejší, v rámci očakávania. Rozhodli sme sa, že skúsime vytvoriť bezdrôtovú sieť na našom notebooku,

a pripojiť VR prilby na našu bezdrôtovú sieť. Prechodom na našu bezdrôtovú sieť sa nám podarilo znížiť čas načítania scény na niekoľko desiatok sekúnd. Tento výsledok nebol ideálny, preto sme sa rozhodli pripraviť testovanie s VR prilbami tak, že scéna už bude spustená a načítaná na VR prilbách, predtým ako ju nasadíme na hlavy účastníkov testovania.

Následne sme skúsili otvoriť scénu na školskom počítači. Načítanie scény bolo rýchlejšie ako na VR prilbách, lebo počítače boli ku školskej sieti pripojené káblom. Avšak aj po načítaní scény bol pohyb v scéne sekavý. V tomto prípade to bolo spôsobené slabým výkonom počítačov. Našťastie nám škola poskytla 3 výkonnejšie počítače, na ktorých bola scéna plynulá. Takže sme testovali na 3 počítačoch a 3 VR prilbách. Avšak keď sa blížil koniec vyučovacej hodiny a videli sme, že sa niektorí žiaci nedostanú ku testovaniu, tak sme im dali možnosť vyskúšať si testovanie na slabšom počítači. Testovanie na slabších počítačoch mohlo skresliť výsledky testovania.

Počas testovania sme si všimli ďalší problém. Keď sme sa do nášho systému prihlásili s používateľským účtom, ktorý ešte nebol použitý, tak sme na ňom mohli vidieť „rozohranú“ scénu. Dôsledky tohto problému sme si neuvedomovali počas testovania, ale po testovaní sme zistili, že nám chýbajú údaje o čase strávenom v scéne, ale aj zozname odpálených prechodov počas používania. Identifikovali sme, že každému používateľovi sa ukazuje každá „rozohraná“ scéna, aj v prípade že ju spustil iný používateľ. To spôsobilo problém, lebo „rozohrané“ scény sme ukončovali, keď sme účastníkom spúšťali scénu. V databáze nám potom chýbali alebo boli nepresné údaje o čase strávenom v scéne a zoznam odpálených prechodov. Na tento problém sme vytvorili na gitlabovom repozitári issue #9¹⁰, aby sa tento problém vyriešil pred ďalším testovaním. Žiaľ kvôli tomuto problému sme stratili časť údajov o účastníkoch testovania, konkrétne o tom ako sa im darilo plniť úlohy.

Testovanie sme začali krátkym predstavením systému a jeho cieľa. Účastníkom sme povedali o čom je scéna, ktorú si pôjdu vyskúšať, a čo v nej budú robiť. Počas vysvetľovania sme im rozdali papiere na získanie informovaných súhlasov pre vyplnenie dotazníka.

Počas testovania niekoľkí účastníci narazili na problém, že sa im stratil nejaký komponent. Tento problém sme im navrhli riešiť obnovením stránky. Za normálnych okolností by im to pomohlo obnoviť scénu, s tým že všetko čo urobili by sa im automaticky vykonalo. Ale kvôli problému s „rozohranými“ scénami,

¹⁰GitLab issue #9 - Dostupné na: <https://git.kpi.fei.tuke.sk/skor-lirkis-projects/LirkisEduVePn/-/issues/9>

ktorý sme spomenul v tejto sekcii, sa im scéna obnovila na začiatok.

Pre VR prilby bola pre každého účastníka potrebná výpomoc s nasadením VR prilby a ovládačov. Na VR prilbe sme im potrebovali nastaviť rozsah šošoviek pre ich zrak. Niektorým účastníkom plno nevyhovovalo žiadne z 3 nastavení šošoviek, ale aj napriek tomu sa rozhodli testovať s VR prilbou. Niektorí účastníci mali problém s ovládaním pomocou ovládačov, ale väčšina účastníkov sa s nimi rýchlo naučila ovládať scénu.

Niektorí účastníci usúdili, že im chýbali presnejšie inštrukcie. Väčšina sa uspokojila, keď sme im naznačili, že v prostredí sú nápomocné plagáty pre inštrukcie na skladanie počítača a uvítacia tabuľa, kde sú inštrukcie na ovládanie. Z pozorovania máme pocit, že niektorí účastníci tieto nápomocné plagáty ignorovali, a snažili sa zistiť čo majú robiť sami.

6.3 Výsledky z dotazníkov

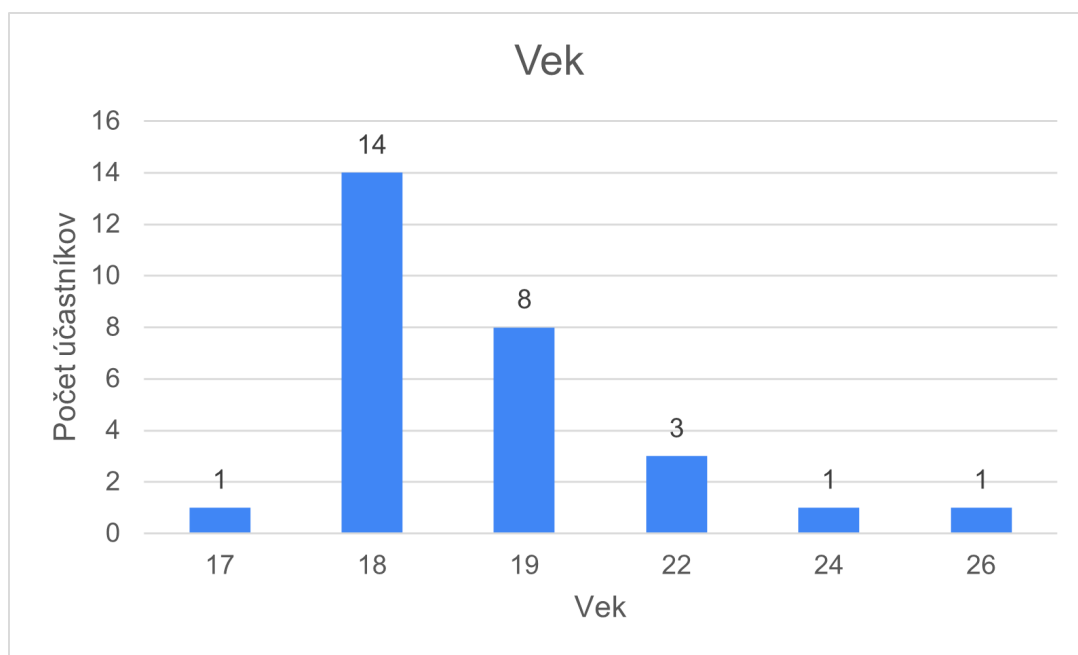
Ku výsledkom z dotazníkov sme chceli pridať aj údaje z databázy systému, ale kvôli problému s „rozohranými“ scénami, ktorý sme rozoberali v predošlej sekcii 6.2, sú tieto údaje neúplné a nepresné. Preto sme sa rozhodli, že údaje z databázy nebudeme vyhodnocovať.

Celkovo nám dotazník vyplnilo 28 účastníkov testovania. Testovania sa zúčastnilo 7 žien a 21 mužov. Z grafu na obrázku 6.1 môžeme vidieť, že účastníci testovania mali vek od 17 do 26 rokov. Väčšina účastníkov testovania mala vek 18 alebo 19 rokov. Testovania sa zúčastnilo 22 žiakov zo Súkromnej strednej odbornej školy ekonomickej KOŠICKÁ AKADÉMIA. Okrem žiakov sa testovania zúčastnili aj 3 ľudia pracujúci mimo IT, 2 ľudia pracujúci v IT a 1 študent z inej školy, ktorý neštuduje v odbore IT.

Skúsenosti s VR malo 22 účastníkov testovania a iba 6 účastníkov nemalo skúsenosti s VR. Toto zistenie nás prekvapilo, lebo sme predpokladali, že väčšina účastníkov testovania nemá skúsenosti s VR. Zaujalo nás aj to, že žiaden z účastníkov neuviedol, že si nie je istý, či má skúsenosti s VR. Vzhľadom na to, že všetci účastníci, ktorý testovali s VR prilbou, mali problém s nastavením šošoviek pre ich zrak a nasadením VR prilby, môžeme predpokladať, že účastníci mali skúsenosti s VR formou VR herne alebo iného testovania podobného nášmu, kde mali pomoc s nasadením VR prilby.

Skúsenosti so skladaním počítača malo 15 účastníkov testovania a 13 účastníkov nemalo skúsenosti so skladaním počítača.

S VR prilbou testovalo 15 účastníkov testovania a 13 účastníkov testovalo na



Obrázok 6.1: Graf znázorňujúci vek účastníkov

počítači. Snažili sme sa o to, aby čo najväčšia skupina účastníkov testovala s VR prilbou, pretože táto forma bola našim hlavným zameraním.

Na otázku „Vadilo ti niečo na systéme?“ sme dostali 3 odpovede, že účastníkom nič nevadilo. Sťažnosti boli hlavne na ťažkosti s chytaním komponentov, náročnosť vkladania menších komponentov. Účastníci by uvítali možnosť priblíženia kamery, schopnosť prisúvať, odťahovať a otáčať komponenty alebo aj možnosť pomalšieho chodenia po prostredí. Niektorí účastníci sa sťažovali aj na problém, že sa im komponenty stratili, ale tento problém sme im navrhli riešiť obnovením stránky, čo nefungovalo kvôli problému s „rozohranými“ scénami spomenutým v sekcii 6.2. Fyzikálna knižnica, ktorú sme použili na simuláciu fyziky komponentov, niekedy spôsobila, že komponenty skákali a lietali po miestnosti aj po malom dotyku.

Na otázku „Páčilo sa ti niečo na systéme?“ sme získali viac pozitívnych odpovedí. Z toho môžeme usúdiť, že aj napriek problémom, ktoré účastníci mali, sa im prostredie páčilo a boli s touto formou výučby spokojní. Naše prostredie hodnotili ako zaujímavé, zábavné a náučné. Niektorí účastníci boli prekvapení, že takáto „hra“ môže byť náučná a aj zábavná. Pochválená bola aj vizuálna stránka počítačových komponentov. Jeden účastník, ktorý testoval prostredie s VR prilbou, bol veľmi spokojný s vizuálnou stránkou prostredia a bol prekvapený, že mu počas testovania neprišlo nevoľno. Zo spätných väzieb z tejto otázky môžeme usúdiť, že o výučbové virtuálne prostredia je záujem zo strany študentov.

Údaje z dotazníka použiteľnosti SUS [22] sme vyhodnotili štandardným spô-

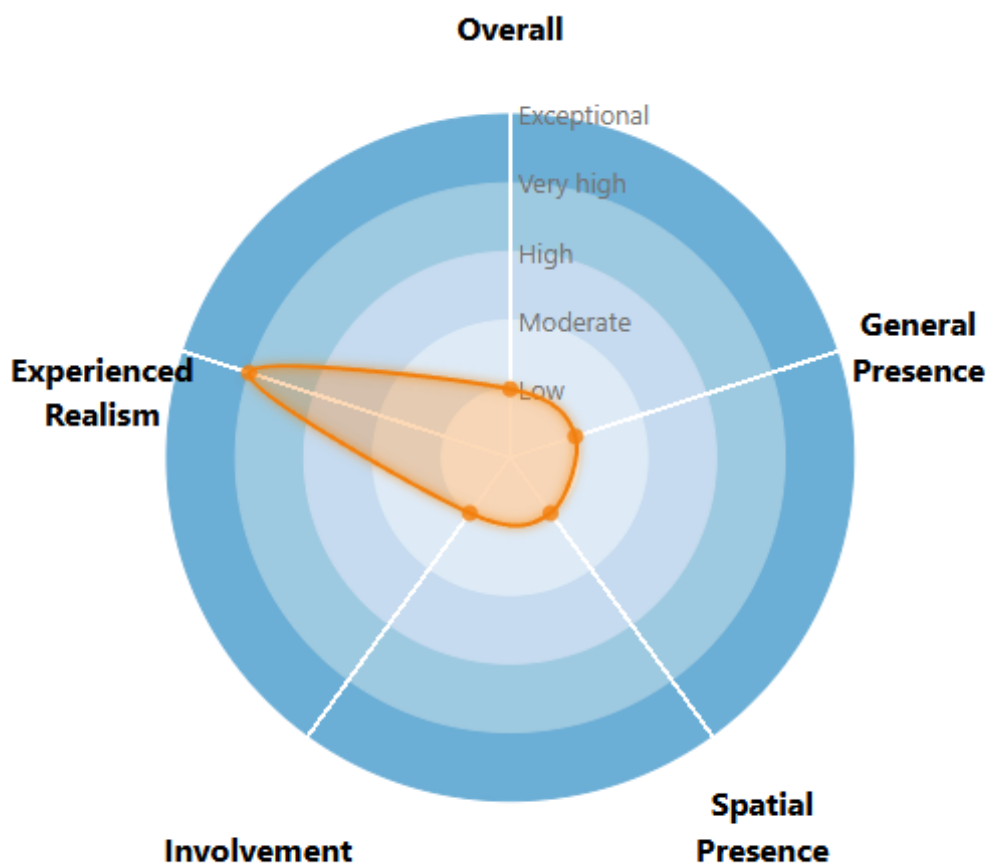
sobom. Dostali sme skóre 72,95. Podľa stupnice hodnotenia z práce [26] vieme toto skóre považovať za dobré. Naše prostredie je vnímané ako použiteľné a účastníci testovania sa s ním vedeli rýchlo naučiť pracovať. Obsahuje určité nedostatky, ktoré by sa mali odstrániť, ale celkovo je prostredie použiteľné.

Dotazník pracovnej záťaže NASA TLX [23] sme sa rozhodli vyhodnotiť formou *Raw TLX*, ktorá sa preukázala ako platná metóda na vyhodnotenie pracovnej záťaže [27]. Týmto spôsobom sme dostali index záťaže 35,48. Podľa stupnice hodnotenia z práce [28] vieme usúdiť, že pracovná záťaž bola mierne vysoká. Predpokladáme, že to môže byť spôsobené tým, že účastníci dostali 2 úlohy, s ktorými menšia polovica nemala žiadne praktické skúsenosti. Problémy s chytaním komponentov mohli tiež prispieť k zvýšenej fyzickej záťaži, hlavne pre tých čo testovali prostredie s VR prilbou. Ak im nejaký komponent spadol na zem, tak sa po neho mohli zohnúť, ak ich nenapadlo použiť ruku s ukazovateľom, ktorá bola určená na chytanie objektov vo väčšej vzdialenosti.

Dotazník prítomnosti vo virtuálnom prostredí IPQ [24] sme vyhodnotili pomocou online nástroja. Tento dotazník vyplňali iba účastníci testovania s VR prilbou. Bývalý študent Tanh z Univerzity v Otagu, ktorý sa venoval výskumu prítomnosti vo virtuálnom prostredí v prácach [29] a [30], vytvoril online nástroj na vyhodnotenie dotazníka IPQ, nazvaný *IPQ Cal*¹¹. Škálu odpovedí sme prepočítali z hodnôt od 1 do 7 na hodnoty od -3 do 3. Pri preklade jednej z otázok sme prevrátili význam otázky, preto sme jej výslednú hodnotu takisto prevrátili. Výsledné dáta sme pripravili do CSV súboru podľa šablóny poskytnutej na stránke IPQ Cal.

Na grafe z obrázku 6.2 môžeme vidieť výsledky z dotazníka IPQ spracované nástrojom IPQ Cal. Nástroj IPQ Cal porovnáva skóre z nášho testovania s predchádzajúcimi štúdiami iných používateľov. Celkové hodnotenie pocitu prítomnosti vo virtuálnom prostredí je v tomto porovnaní nízke. Pre triedy pocitu všeobecnej prítomnosti, priestorovej prítomnosti, zapojenia do aktivity boli hodnotenia nízke. Ale pre triedu pocitu zažitého realizmu boli hodnotenia veľmi vysoké. Z tohto hodnotenia usudzujeme, že prostredie účastníci vnímali ako vizuálne realistické a uveriteľné. Ale účastníci nemali silný pocit, že sú priamo prítomní vo virtuálnom prostredí a necítili sa byť silne zapojení do úloh vo virtuálnom prostredí. Toto hodnotenie môže byť vhodné pre náš systém, lebo vieme konštatovať, že si žiaci uvedomovali, že sú prítomní v škole a nie vo virtuálnom prostredí, aj napriek tomu, že prostredie pôsobilo realisticky. Negatívnou stránkou je hodnotenie zapojenia do aktivity, ktoré je nízke. To by mohlo naznačovať, že účastníci

¹¹IPQ Cal - dostupné na: <https://hci.otago.ac.nz/ipqcal/>



Obrázok 6.2: Graf znázorňujúci výsledky z dotazníka IPQ - poskytnutý nástrojom IPQ Cal

nepovažovali úlohy za dôležitú časť prostredia.

Dotazník nevoľnosti vo virtuálnom prostredí VRSQ [25] sme vyhodnotili vypočítaním priemeru a medianu pre jednotlivé typy nevoľnosti. Účastníci zadávali hodnotenie od 0 do 3 podľa toho ako silný pocit nevoľnosti mali. Význam jednotlivých hodnôt je znázornený v tabuľke 6.1. V tabuľke 6.2 sú znázornené spracované výsledky z dotazníka VRSQ. Z výsledkov môžeme usúdiť, že účastníci testovania mali najväčšie problémy s rozmazaným videním, ťažkosťami so zaostrením a námahou očí. Viac ako polovica účastníkov testovania aspoň mierne pociťovala tieto typy nevoľnosti. Tieto výsledky zodpovedajú našim počiatočným predpokladom, kde sme očakávali, že približne polovica žiakov bude mať problémy s rozmazaným videním vo VR prilbe. Takéto výsledky môžu naznačovať problém s použitými VR prilbami, ktoré nemajú dostatočne presné nastavenie šošoviek, ale poskytujú iba 3 nastavenia, ktoré nemusia vyhovovať každému účastníkovi. Vizuálne problémy s VR prilbami mohli negatívne ovplyvniť hodnotenie dotazníka prítomnosti v prostredí (IPQ). Žiadny typ nevoľnosti nemal

Stupeň	Hodnotenie
0	žiadny(a)
1	mierny(a)
2	stredný(á)
3	vážny(a)

Tabuľka 6.1: Tabuľka znázorňujúca stupnicu hodnotenia pre dotazník VRSQ

Typ nevoľnosti	Priemer	Median
Všeobecný diskomfort	0,53	0
Únava	0,53	0
Námaha očí	0,87	1
Ťažkosti so zaostrením	0,87	1
Bolesť hlavy	0,4	0
Tlak v hlave	0,33	0
Rozmazané videnie	1,07	1
Závraty pri zatvorených očiach	0,27	0
Vertigo (točenie hlavy)	0,33	0

Tabuľka 6.2: Tabuľka znázorňujúca výsledky z dotazníka VRSQ

priemer blízky stupňu 2 (stredný) alebo 3 (vážny). To naznačuje, že virtuálne prostredie bolo z hľadiska nevoľnosti relatívne dobre tolerované.

7 Záver

Hlavným cieľom tejto bakalárskej práce bolo vytvoriť nové virtuálne výučbové prostredie pre projekt *LirkisEduVePn*. Vytvorené prostredie je zamerané na skladanie počítača a výmenu počítačových komponentov. Vytvorili sme virtuálne prostredie podobné kancelárii, kde je možné vykonávať rôzne úlohy typu skladanie počítača a výmena jedného alebo viacerých komponentov. Úlohy v prostredí sú popísané scenárom vo forme Petriho siete. Ako inšpiráciu sme predpripravili 4 rôzne scenáre, ktoré môžu poslúžiť ako základ pre tvorcu nových scenárov.

Na začiatku práce sme potrebovali analyzovať aktuálny stav projektu, aby sme zistili aké funkcionality sú už implementované a aké sú potrebné pre naše prostredie. Zistili sme, že v projekte už existuje správa scenárov pre úpravy úloh v prostredí. Ku scenárom patria aj jazykové súbory, ktoré sú určené na preklad textov v prostredí. Doterajšie vytvorené prostredia sú ovládateľné pomocou počítača, VR prilby a mobilných zariadení. Všetky spomenuté funkcionality sme pre naše prostredie zachovali.

Vykonalí sme prieskum metódou systematického prehľadu literatúry, aby sme zistili či už existuje virtuálne výučbové prostredia zamerané na skladanie počítača. Zistili sme, že takéto prostredia už existujú a sú vytvorené pre podporu rôznych zariadení. Našli sme prostredia vytvorené pre VR, AR a aj pre klasické počítače. Tieto prostredia používali aj rôzne vstupné zariadenia ako sú myš, klávesnica, VR ovládače, sledovanie rúk v rôznych formách. Prostredia mali fotorealistické ale aj jednoduché modely počítačových komponentov, dokonca sme našli aj 2D prostredie. Týmto prieskumom sme získali inšpiráciu pre naše prostredie a zistili sme akým chybám sa máme vyhnúť pri návrhu a implementácii.

Navrhli sme naše prostredie pomocou voľne dostupných 3D modelov kancelárskeho prostredia a počítačových komponentov. Modely sme upravovali, aby mali realistické rozmery vo virtuálnom prostredí. Do prostredia sme pridali aj nápomocné plagáty, ktoré popisujú ako sa prostredie ovláda, ale aj ako sa má skladať počítač. Scenár sme navrhli vo forme Petriho siete, pričom sme sa snažili o možnosť jednoduchšej úpravy scenárov pre učiteľov a iných budúcich tvorcov.

Prepojili sme scenár s prostredím tak, že sa počiatočný stav komponentov v prostredí vie upravovať pomocou scenáru.

V implementačnej časti sme riešili hlavne spájanie, odpájanie a kolízie komponentov počítača. Rozhodli sme sa použiť knižnicu pre realistickú fyziku v našom prostredí.

Pripravili sme prieskum pre žiakov 3. a 4. ročníka strednej školy študujúcich informatický odbor. V prostredí dostali úlohy poskladať počítač a vymeniť grafickú kartu. Kombináciou viacerých štandardizovaných dotazníkov sme vyzbierali spätnú väzbu od žiakov. Na základe analýzy výsledkov sme zistili, že naše prostredie je dobre použiteľné pre výučbu skladania počítača. Žiaci prostredie popísali ako zaujímavé, zábavné a náučné. Prostredie podľa žiakov pôsobilo veľmi realisticky a boli s ním spokojní. Testovanie v nich nevyvolalo žiadne vážne pocity nevoľnosti a ani diskomfortu. Na druhej strane sme zistili, že pracovná záťaž bola pre žiakov mierne vysoká. V prostredí nepocítili pocit prítomnosti a ani pocit zapojenia do aktivity. V priemere mali mierne ťažkosti so zaostrením, rozmazaným videním a námahou očí, čo mohli spôsobiť použité VR prilby.

Žiaci identifikovali niekoľko chýb v prostredí, ktoré im bránili v dokončení úloh v prostredí. Poukázali taktiež na niekoľko funkcionalít, ktoré by im uľahčili prácu v prostredí. Nad väčšinou týchto funkcionalít sme premýšľali aj my, no neboli implementované kvôli časovým obmedzeniam. Aj napriek chybám a nedostatkom v prostredí boli žiaci s prostredím spokojní. Preto usudzujeme, že prostredie je vytvorené správne a je použiteľné pre výučbu skladania počítača.

Zoznam použitej literatúry

1. KAŠELA, Adam. *Výučbové scenáre v rozšírenej realite využívajúce procesné grafy*. Košice, 2022. Technická univerzita v Košiciach, Fakulta elektrotechniky a informatiky. 65s. Vedúci práce: Ing. Štefan Korečko, PhD.
2. DEMIANENKO, Dmytro. *Spracovanie údajov z virtuálnych výučbových prostredí*. Košice, 2023. Technická univerzita v Košiciach, Fakulta elektrotechniky a informatiky. 65s. Vedúci práce: Ing. Štefan Korečko, PhD.
3. PISARČÍK, Lukáš. *Scenárom riadené výučbové prostredia vo virtuálnej realite*. Košice, 2023. Technická univerzita v Košiciach, Fakulta elektrotechniky a informatiky. 47s. Vedúci práce: Ing. Štefan Korečko, PhD.
4. RUDY, Juraj. *Systém virtuálnych výučbových prostredí riadených procesnými grafmi*. Košice, 2024. Technická univerzita v Košiciach, Fakulta elektrotechniky a informatiky. 59s. Vedúci práce: Ing. Štefan Korečko, PhD.
5. ZSIGÓ, Kristián. *Rozšírenia a ukážková scéna pre systém virtuálnych výučbových prostredí*. Košice, 2025. Technická univerzita v Košiciach, Fakulta elektrotechniky a informatiky. 75s. Vedúci práce: Ing. Štefan Korečko, PhD.
6. GUERRA, Vania; KUHNT, Beate; BLÖCHLIGER, Ivo. Informatics at school-worldwide. *An international exploratory study about informatics as a subject at different school levels*. 2012.
7. DEDE, Chris; MISHRA, Punya; VOOGT, Joke. Working group 6: Advancing computational thinking in 21st century learning. In: *EDUsummIT 2013, International summit on ict in education*. 2013.
8. A.G., Abdullah; A.B.D., Nandiyanto; I., Widiaty; A.A., Danuwijaya; C.U., Abdullah (ed.). 4th Annual Applied Science and Engineering Conference - Preface. In: Institute of Physics Publishing, 2019, zv. 1402. Č. 1. ISSN 17426588. Dostupné tiež z: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85078823139&partnerID=40&md5=4706b59288324a1aa91a696b23e71313>. Cited by: 0.

9. TÜZÜN, Hakan. Blending video games with learning: Issues and challenges with classroom implementations in the Turkish context. *British Journal of Educational Technology*. 2007, roč. 38, č. 3, s. 465–477. Dostupné z doi: 10.1111/j.1467-8535.2007.00710.x. Cited by: 60; All Open Access, Green Open Access.
10. CHECA, David; MIGUEL-ALONSO, Ines; BUSTILLO, Andres. Immersive virtual-reality computer-assembly serious game to enhance autonomous learning. *Virtual Reality*. 2023, roč. 27, č. 4, s. 3301–3318. ISSN 13594338. Dostupné z doi: 10.1007/s10055-021-00607-1. Cited by: 46; All Open Access, Hybrid Gold Open Access.
11. TONG, Yanping; XIE, Fu; ZHENG, Xiangwei; WEI, Yi. Design and application of virtual training system for computer hardware assembly. In: Association for Computing Machinery, 2020, s. 146–150. ISBN 978-145038787-3. Dostupné z doi: 10.1145/3425329.3425365. Cited by: 3.
12. GEETHA, S.; ADITYA, G.; CHETAN REDDY, M.; NISCHITH, G. Human Interaction in Virtual and Mixed Reality Through Hand Tracking. In: Institute of Electrical a Electronics Engineers Inc., 2024. ISBN 979-835038592-2. Dostupné z doi: 10.1109/CONECCT62155.2024.10677239. Cited by: 0.
13. HAOXIN, Wu; XIAOWEI, An; QINQIN, Xian; NONGLIANG, Sun. Immersive Experience-based Cooperative Interaction for Computer Assembly Guidance. In: Institute of Electrical a Electronics Engineers Inc., 2021, s. 443–448. ISBN 978-172818319-0. Dostupné z doi: 10.1109/ICCECE51280.2021.9342391. Cited by: 0.
14. CRIOLLO-C, Santiago; GUERRERO-ARIAS, Andrea; UZCATEGUI, Jose Enrique Cerezo; ARIF, Yunifa Miftachul; FORTUNA, Aprilla; PRASETYA, Febri; LUJAN-MORA, Sergio. Improving Higher Education With the Use of Mobile Augmented Reality (MAR): A Case Study. *IEEE Access*. 2024, roč. 12, s. 139003–139017. ISSN 21693536. Dostupné z doi: 10.1109/ACCESS.2024.3465833. Cited by: 0; All Open Access, Gold Open Access.
15. TAYAG, Marlon; RONIE, Bituin; MARVIN, Reyes. Leap Motion Controller Enabled Simulations of Personal Computer Assembly Programs. In: Association for Computing Machinery, 2021, s. 17–21. ISBN 978-145038895-5. Dostupné z doi: 10.1145/3451471.3451474. Cited by: 1.
16. RUSCANU, Ana-Maria; CIUPE, Aurelia; MEZA, Serban Nicolae. 'arPcTECHture' - a gamified educational 3D virtual world for introductory concepts in

- computer architecture. In: M., Jemni; I., Kallel; A., Akkari (ed.). IEEE Computer Society, 2022, zv. 2022-March, s. 1437–1442. ISBN 978-166544434-7. ISSN 21659559. Dostupné z DOI: 10.1109/EDUCON52537.2022.9766679. Cited by: 1.
17. LU, Mengjiao; SHAO, Xiuting; SONG, Ziwen; CHEN, Hengbiao. The Design and Implementation of Virtual Simulation System for Computer Assembly and Maintenance Course in Vocational School. In: Institute of Electrical and Electronics Engineers Inc., 2023, s. 108–113. ISBN 979-835031915-6. Dostupné z DOI: 10.1109/ITME60234.2023.00032. Cited by: 0.
18. MA, Shi-Xia; GUO, Zu-Hua; LIU, Dan. The application of VRML technology in the virtual laboratory of computer assembly. In: 2009, s. 328–331. ISBN 978-142444519-6. Dostupné z DOI: 10.1109/ICCSIT.2009.5234935. Cited by: 0.
19. XU, Hongzhen; LU, Ling; SONG, Wenlin; ZHOU, Shumin. The distance education system based on VRML. In: 2008, zv. 5, s. 879–881. ISBN 978-076953336-0. Dostupné z DOI: 10.1109/CSSE.2008.10. Cited by: 0.
20. HOU, Huei-Tse; LI, Ming-Chaun. Evaluating multiple aspects of a digital educational problem-solving-based adventure game. *Computers in Human Behavior*. 2014, roč. 30, s. 29–38. ISSN 07475632. Dostupné z DOI: 10.1016/j.chb.2013.07.052. Cited by: 120.
21. TRINADI, A.; GERHANA, Y.A.; SYARIPUDIN, U. The implementation of the depth-first search algorithm on learning computer assembly 3D simulations in vocational education. In: A.G., Abdullah; A.B.D., Nandiyanto; I., Widiaty; A.A., Danuwijaya; C.U., Abdullah (ed.). Institute of Physics Publishing, 2019, zv. 1402. Č. 7. ISSN 17426588. Dostupné z DOI: 10.1088/1742-6596/1402/7/077002. Cited by: 1; All Open Access, Gold Open Access.
22. AND, James R. Lewis. The System Usability Scale: Past, Present, and Future. *International Journal of Human - Computer Interaction*. 2018, roč. 34, č. 7, s. 577–590. Dostupné z DOI: 10.1080/10447318.2018.1455307.
23. HART, Sandra G.; STAVELAND, Lowell E. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In: HANCOCK, Peter A.; MESHKATI, Najmedin (ed.). *Human Mental Workload*. North-Holland, 1988, zv. 52, s. 139–183. Advances in Psychology. ISSN 0166-4115. Dostupné z DOI: [https://doi.org/10.1016/S0166-4115\(08\)62386-9](https://doi.org/10.1016/S0166-4115(08)62386-9).

24. SCHUBERT, Thomas; FRIEDMANN, Frank; REGENBRECHT, Holger. The Experience of Presence: Factor Analytic Insights. *Presence*. 2001, roč. 10, s. 266–281. Dostupné z doi: 10.1162/105474601300343603.
25. KIM, Hyun K.; PARK, Jaehyun; CHOI, Yeongcheol; CHOE, Mungyeong. Virtual reality sickness questionnaire (VRSQ): Motion sickness measurement index in a virtual reality environment. *Applied Ergonomics*. 2018, roč. 69, s. 66–73. ISSN 0003-6870. Dostupné z doi: <https://doi.org/10.1016/j.apergo.2017.12.016>.
26. BANGOR, Aaron; KORTUM, Phil; MILLER, James. Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *J. Usability Stud.* 2009, roč. 4, s. 114–123.
27. SAID, Sadiq; GOZDZIK, Malgorzata; ROCHE, Tadzio; BRAUN, Julia; RÖSSLER, Julian; KASERER, Alexander; SPAHN, Donat; NÖTHIGER, Christoph; TSCHOLL, David. Validation of the Raw National Aeronautics and Space Administration Task Load Index (NASA-TLX) Questionnaire to Assess Perceived Workload in Patient Monitoring Tasks: Pooled Analysis Study Using Mixed Models. *Journal of Medical Internet Research*. 2020, roč. 22.
28. PRABASWARI, Atyanti; BASUMERDA, Chancard; UTOMO, Bagus. The Mental Workload Analysis of Staff in Study Program of Private Educational Organization. *IOP Conference Series: Materials Science and Engineering*. 2019, roč. 528, s. 012018. Dostupné z doi: 10.1088/1757-899X/528/1/012018.
29. TRAN, Thanh Quang; LANGLOTZ, Tobias; YOUNG, Jacob; SCHUBERT, Thomas W.; REGENBRECHT, Holger. Classifying Presence Scores: Insights and Analysis from Two Decades of the Igroup Presence Questionnaire (IPQ). *ACM Trans. Comput.-Hum. Interact.* 2024, roč. 31, č. 5. ISSN 1073-0516. Dostupné z doi: 10.1145/3689046.
30. TRAN, Thanh Quang; LANGLOTZ, Tobias; REGENBRECHT, Holger. A Survey On Measuring Presence in Mixed Reality. In: *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. Honolulu, HI, USA: Association for Computing Machinery, 2024. CHI '24. ISBN 9798400703300. Dostupné z doi: 10.1145/3613904.3642383.

Zoznam príloh

Príloha A Celý model Petriho siete

Príloha B Používateľská príručka

Príloha C Systémová príručka

Príloha D CD médium – záverečná práca v elektronickej podobe,

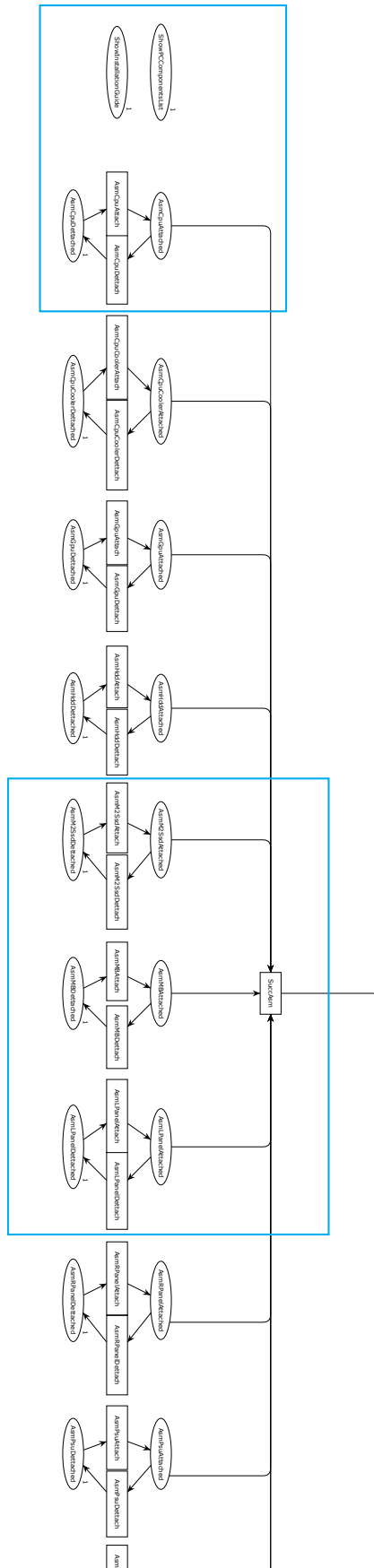
Príloha E Repozitár výsledného projektu je dostupný na adrese

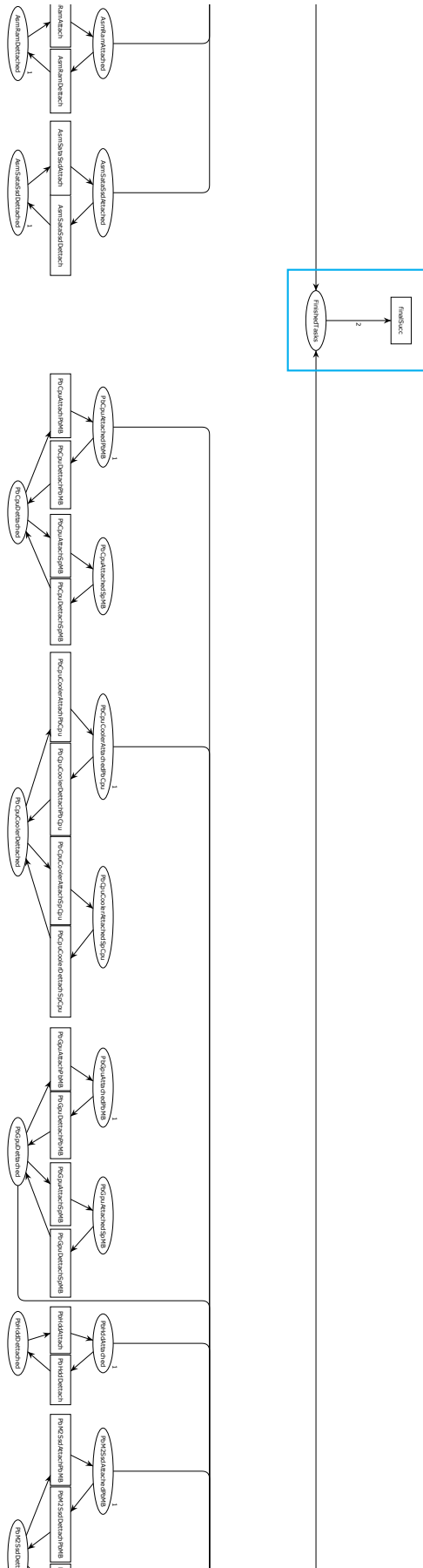
<https://git.kpi.fei.tuke.sk/skor-lirkis-projects/LirkisEduVePn>

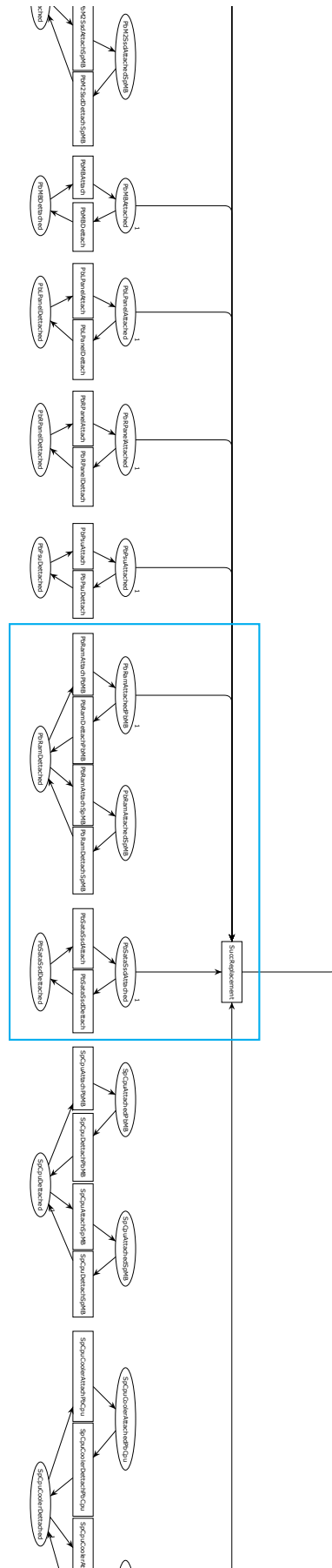
A Celý model Petriho siete

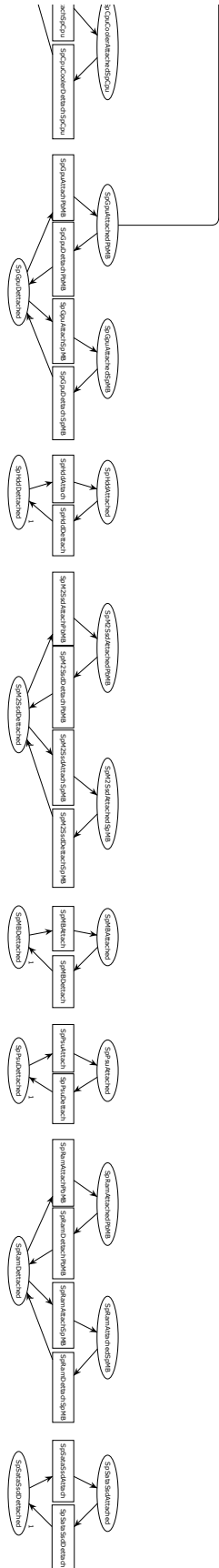
Na nasledujúcich stránkach je celá Petriho sieť pre testovací scenár. Časti označené modrými obdlžníkmi sú tie, ktoré boli zobrazené v sekcii 4.3 v obrázkoch 4.11, 4.12, 4.13 a 4.14. Vzhľadom na veľkosť Petriho siete je odporúčané si ju pozrieť v elektronickej podobe, s možnosťou priblíženia.

Môžete si všimnúť, že niektoré časti siete nie sú pripojené ku finálnym prechodom *SuccAsm* a *SuccReplacement*. Tieto časti patria ku úlohe na výmenu komponentov. Oddelené časti nie sú potrebné pre našu úlohu v tomto scenári. Keďže v tomto scenári chceme vymeniť iba grafickú kartu, tak nás ostatné náhradné komponenty nezaujímajú. Ak by ste tento testovací scenár porovnali s ťažkým scenárom, v ktorom je potrebné vymeniť základnú dosku, tak by ste si všimli, že iné časti siete sú prepojené a iné sú oddelené.









B Používateľská príručka

Používateľská príručka, rovnako ako práca, priamo nadväzuje na práce Adama Kašelu [1], Dmytra Demianenka [2], Lukáša Pisarčíka [3] a Juraja Rudyho [4] a niektoré aspekty používateľskej príručky, ktoré sa našou prácou nijak nemenili sme ponechali v pôvodnom stave.

B.1 Funkcia programu

Táto práca sa zameriava na systém *LirkisEduVePn*. Systém je určený na virtuálne výučbové prostredia ovládateľné pomocou počítača, VR prilby alebo mobilného zariadenia. Poskytuje učiteľom dohľad nad úlohami a ponúka administratívne funkcie. Cieľom systému je zhromažďovať údaje o používateľoch počas plnenia úloh vo virtuálnych prostrediach a poskytovať podrobné štatistiky o splnených úlohách, popri tom ako sa žiaci vzdelávajú vo virtuálnych prostrediach. Integrácia možností analýzy údajov umožňuje získať prehľad v reálnom čase a personalizovanú spätnú väzbu pre používateľov, čo zlepšuje výsledky vzdelávania.

My sme do programu pridali nové virtuálne prostredie zamerané na výučbu skladania počítačov. Jej cieľom je aby sa používatelia naučili vizuálne rozpoznávať jednotlivé počítačové komponenty a vedeli ich správne poskladať do funkčného počítača. Nezameriavame sa na to, aby sa používatelia naučili každý jeden krok skladania počítača, ale skôr chceme aby získal prehľad o tom kde by sa mali jednotlivé komponenty vložiť.

B.2 Inštalácia programu

V tejto kapitole sa rozoberá obsah produktu, po ktorom nasleduje špecifikácia technických a softvérových požiadaviek. Postup inštalácie produktu je opísaný z pohľadu používateľa (administrátora aj bežného používateľa).

B.2.1 Požiadavky na technické prostriedky

Na spustenie projektu musia byť splnené tieto minimálne hardvérové požiadavky pre WebGL, a to:

- 64-bitový operačný systém
- 4 GB pamäte RAM
- Integrovaná grafika s podporou WebGL 2.0 OpenGL ES 3.0
- Dvojjadrový procesor Intel/AMD
- Najnovšia verzia prehliadača Chrome, Firefox alebo Opera

B.2.2 Spustenie projektu

Táto časť opisuje kroky potrebné na otvorenie projektu vo webovom prehliadači pre bežných používateľov alebo na nasadenie projektu na server pre administrátorov.

Otvorenie projektu pre používateľov

Každý používateľ, ktorý má prístup k sieti VPN TUKE alebo k sieti Wi-Fi TUKE, môže tento projekt otvoriť v ľubovoľnom webovom prehliadači na adrese <https://147.232.205.222:4200>, kde sa potom môže zaregistrovať alebo prihlásiť. V prípade, že je tento systém na inom serveri ako je TUKE server s IP adresou 147.232.205.222, je potrebné zmeniť IP adresu v URL adrese.

Nasadenie aplikácie pre administrátorov

Ak administrátor potrebuje nasadiť projekt na server, môže toto urobiť pomocou nástroja Docker¹².

Po nainštalovaní nástroja Docker stačí na úspešné spustenie projektu len niekoľko krokov. Najprv je potrebné vytvoriť prázdny priečinok na serveri s dvoma súborami: `docker-compose.yaml` a `servers.json`.

Súbor `docker-compose.yaml` slúži na definovanie služieb, sietí a zväzkov potrebných pre nastavenie nástroja Docker. Umožňuje konfigurovať a spravovať viacero kontajnerov ako jednu aplikáciu. Príklad základného súboru `docker-compose.yaml` je vo výpise B.1.

¹²Docker je open-source platforma, ktorá umožňuje kontajnerizáciu aplikácií a poskytuje izolované a prenosné prostredie. Viac informácií nájdete na adrese: <https://www.docker.com/>

Výpis B.1: Príklad základného súboru docker-compose.yaml

```
version: '3.8'

services:

  db-postgres:
    image: "postgres:15.1"
    container_name: lirkis-database
    volumes:
      - lirkis-data:/var/lib/postgresql/data
    ports:
      - "5432:5432"
    environment:
      - POSTGRES_DB=postgres
      - POSTGRES_USER=postgres
      - POSTGRES_PASSWORD=postgres

  pgadmin:
    image: dpage/pgadmin4:7.1
    container_name: lirkis-pgadmin
    depends_on:
      - db-postgres
    ports:
      - "3000:80"
    environment:
      - PGADMIN_DEFAULT_EMAIL=admin@admin.com
      - PGADMIN_DEFAULT_PASSWORD=admin
    volumes:
      - ./servers.json:/pgadmin4/servers.json

  spring:
    image: yuray/spring:deploy_06052024
    container_name: lirkis-service
    ports:
      - "8443:8443"
    links:
      - db-postgres

  angular:
```

```

image: zsigipajka/angular:deploy_v3.0
container_name: lirkis-ui
ports:
  - "4200:443"
links:
  - spring

volumes:
  lirkis-data:

```

V tomto súbore máme definované štyri služby: db-postgres, pgadmin, spring a angular.

Služba „db-postgres“ používa oficiálny obraz „postgres“, nastavuje premenné prostredia pre predvoleného používateľa, heslo a názov databázy a pripája adresár ./data ako zväzok na uchovávanie údajov databázy. Databáza bude prístupná na porte 5432.

Služba „spring“ je prevzatá z obrazu „yuray/spring:deploy_06052024“, ktorý bol zostavený a nahraný na Docker Hub Jurajom Rudym [4], a vystavuje port 8443 na hostiteľovi, pričom ho mapuje na port 8443 v kontajneri. Táto služba funguje ako server.

Služba „angular“ je prevzatá z obrazu „zsigipajka/angular:deploy_v3.0“, ktorý bol zostavený a nahraný na Docker Hub Kristiánom Zsigom [5], a vystavuje port 4200 na hostiteľovi a mapuje ho na port 443 v rámci kontajnera. Táto služba funguje ako klient.

Služba „pgadmin“ používa obraz „dpage/pgadmin4“, nastavuje premenné prostredia pre predvolený e-mail a heslo používateľa admin a vystavuje port 3000 na hostiteľovi, pričom ho mapuje na port 80 v rámci kontajnera. Okrem toho pripája súbor ./servers.json ako zväzok, ktorý tiež musí byť nakonfigurovaný.

Na výpise B.2 je uvedený súbor „servers.json“, ktorý používa služba „pgadmin“ na konfiguráciu pripojenia k serveru PostgreSQL.

Výpis B.2: Príklad súboru servers.json

```

{
  "Servers": {
    "1": {
      "Name": "Local",
      "Group": "Servers",
      "Host": "db-postgres",
      "Port": 5432,

```

```

    "MaintenanceDB": "postgres",
    "Username": "postgres",
    "SSLMode": "prefer",
    "SSLCert": "<STORAGE_DIR>/.postgresql/postgresql.crt",
    "SSLKey": "<STORAGE_DIR>/.postgresql/postgresql.key",
    "SSLCompression": 0,
    "Timeout": 10,
    "UseSSHTunnel": 0,
    "TunnelPort": "22",
    "TunnelAuthentication": 0
  }
}
}

```

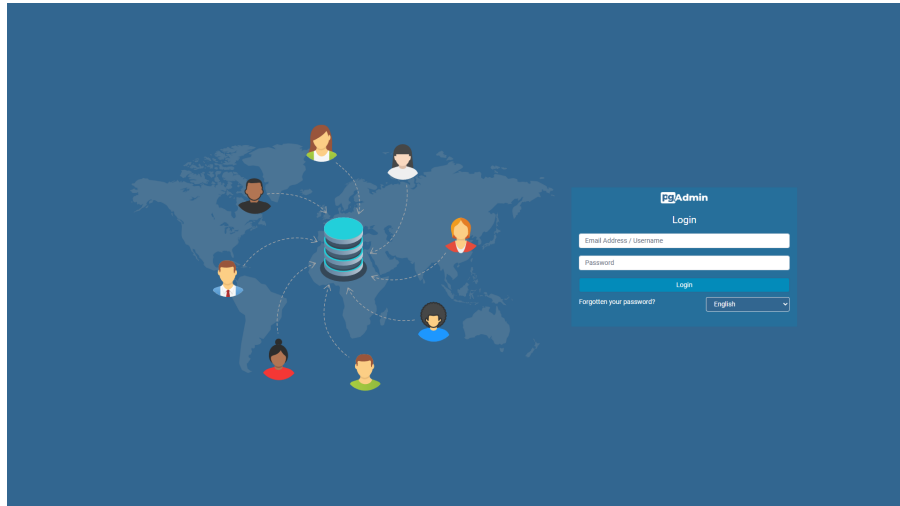
Po vytvorení dvoch vyššie uvedených súborov stačí na spustenie programu napísať príkaz „docker compose up“, ak používate systém Windows, alebo „docker-compose up“, ak používate systém Linux.

Po spustení programu môžete prejsť na stránku <https://147.232.205.222:4200>, ktorá bude mať všetky funkcie rovnaké ako v používateľskom režime. V prípade, že je tento systém na inom serveri ako je TUKE server s IP adresou 147.232.205.222, je potrebné zmeniť IP adresu v URL adrese.

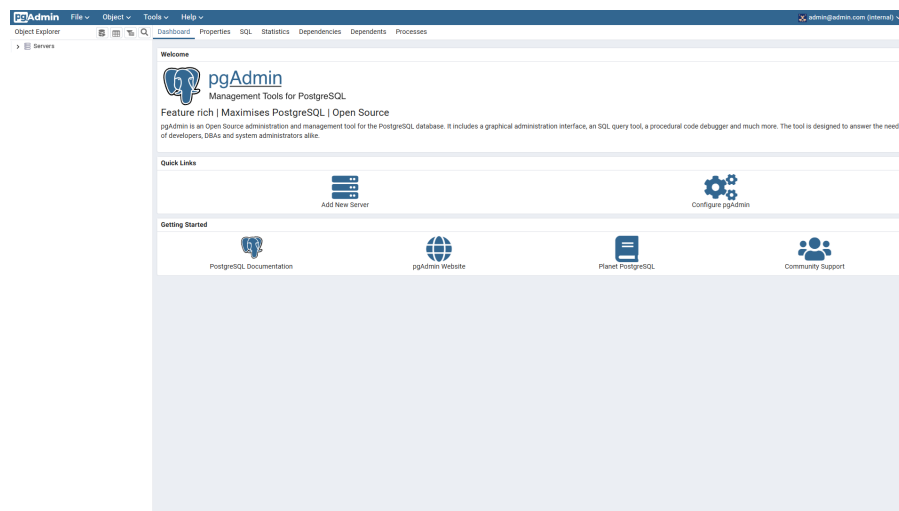
Veľmi dôležitým aspektom je aj pridanie správcu do systému. Keďže pri písaní práce nebolo možné zabezpečiť, aby sa používateľ s rolou administrátora pri vytváraní databázy pomocou skriptu SQL pridával automaticky, je potrebné pridať tohto používateľa ručne. Ak to chcete urobiť, musíte najprv prejsť na stránku <http://147.232.205.222:3000>, ktorá je zodpovedná za grafický panel databázy. V prípade, že je tento systém na inom serveri ako je TUKE server s IP adresou 147.232.205.222, je potrebné zmeniť IP adresu v URL adrese. Na obrázku B.1 je možné vidieť prihlasovacie okno.

Ak sa chcete prihlásiť, musíte zadať údaje uvedené v súbore „docker-compose.yml“ v službe „pgAdmin“. V našom prípade je to e-mail „admin@admin.com“ a heslo „admin“. Po overení sa otvorí ovládací panel. Na obrázku B.2 vľavo hore v rohu vidíte počet registrovaných serverov, v ktorých bude naša databáza. Ak sa pokúsite otvoriť našu databázu, zobrazí sa okno s požiadavkou na zadanie hesla na prístup do databázy. Heslo je „postgres“. Po zadaní hesla sa otvorí naša databáza, ktorú môžete vidieť na obrázku B.3.

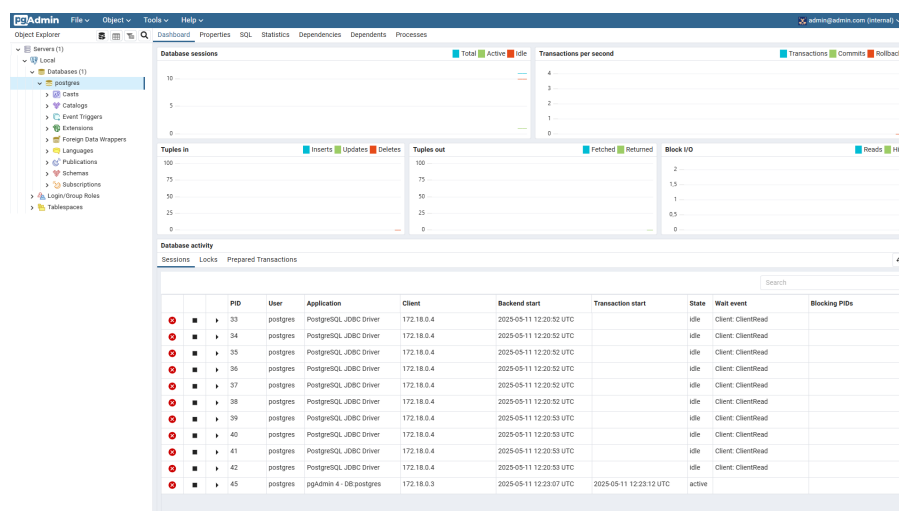
Po otvorení databázy kliknite na tlačidlo „Query Tool“ alebo použite klávesovú skratku „Alt + Shift + Q“, čím otvoríte editor dotazov SQL. Po otvorení editora zadajte dotaz z výpisu B.3, ktorý pridá nového používateľa s administrátorskými



Obrázok B.1: Prihlasovacia obrazovka do pgAdmin



Obrázok B.2: Okno so servermi v pgAdmin



Obrázok B.3: Databáza v pgAdmin

právami.

Výpis B.3: SQL dotaz na pridanie používateľa s administrátorskými právami

```
INSERT INTO users(
    email , firstname , is_enabled ,
    lastname , username , password , role
)
VALUES (
    'admin@admin', 'admin', true , 'admin', 'admin',
    '$2a$12$3HMoKSwt7wbRayzLmoApieJTfvP0oqtc5ZTYmsPyo4PSJoy9TeDmW',
    'ADMIN'
);
```

Po pridaní používateľa s administrátorskými právami sa môžete autentifikovať na serveri pomocou používateľského mena „admin@admin“ a hesla „admin“, ktoré potom môžete zmeniť. Po autentifikácii bude môcť administrátor využívať všetky funkcie systému.

Študenti a iný bežný používatelia sa môžu registrovať do systému pomocou registračného formulára. Ale ak by ste im chceli uľahčiť prácu a nechcete, aby sa registrovali sami, môžete im vytvoriť účty pomocou skriptu SQL z výpisu B.4. Tento skript obsahuje 3 dotazy, ktoré vytvoria 5 používateľov s rolou študenta, skupinu pomenovanú „Participants“ a priradia všetkých 5 používateľov do tejto skupiny. Do systému sa môžu prihlásiť pomocou používateľského mena „p00@gmail.com“ až „p04@gmail.com“ a hesla „Password1“. Tento skript môžete rozšíriť o ďalších používateľov, ak je to potrebné.

Výpis B.4: SQL dotazy na vytvorenie účtov pre testovanie

```
INSERT INTO users (
    email, firstname, is_enabled,
    lastname, username, password, role
)
VALUES (
    'p00@gmail.com', 'Participant', true, '00', 'P00',
    '$2a$10$jMddI4QXpWahvCdjLTzRp0HEZiUcM3boLJ0j3V0jXqi4CeaG6yXLq',
    'STUDENT'
), (
    'p01@gmail.com', 'Participant', true, '01', 'P01',
    '$2a$10$jMddI4QXpWahvCdjLTzRp0HEZiUcM3boLJ0j3V0jXqi4CeaG6yXLq',
    'STUDENT'
), (
```



```

    'p02@gmail.com', 'Participant', true, '02', 'P02',
    '$2a$10$jMddI4QXpWahvCdjLTzRp0HEZiUcM3boLJ0j3V0jXqi4CeaG6yXLq',
    'STUDENT'
), (
    'p03@gmail.com', 'Participant', true, '03', 'P03',
    '$2a$10$jMddI4QXpWahvCdjLTzRp0HEZiUcM3boLJ0j3V0jXqi4CeaG6yXLq',
    'STUDENT'
), (
    'p04@gmail.com', 'Participant', true, '04', 'P04',
    '$2a$10$jMddI4QXpWahvCdjLTzRp0HEZiUcM3boLJ0j3V0jXqi4CeaG6yXLq',
    'STUDENT'
);

```

```
INSERT INTO groups (name) VALUES ('Participants');
```

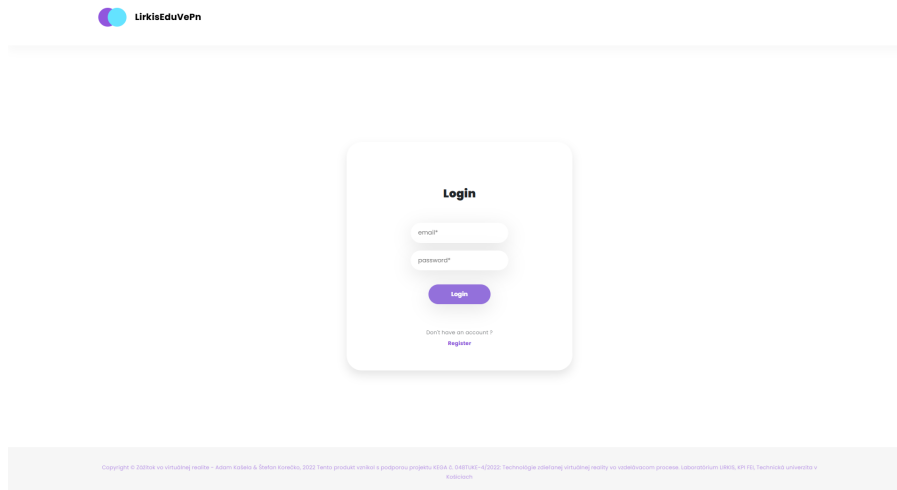
```
INSERT INTO users_groups (user_id, group_id)
SELECT u.id, g.id
FROM users u, groups g
WHERE g.name = 'Participants'
AND u.firstname = 'Participant';
```

B.3 Použitie programu

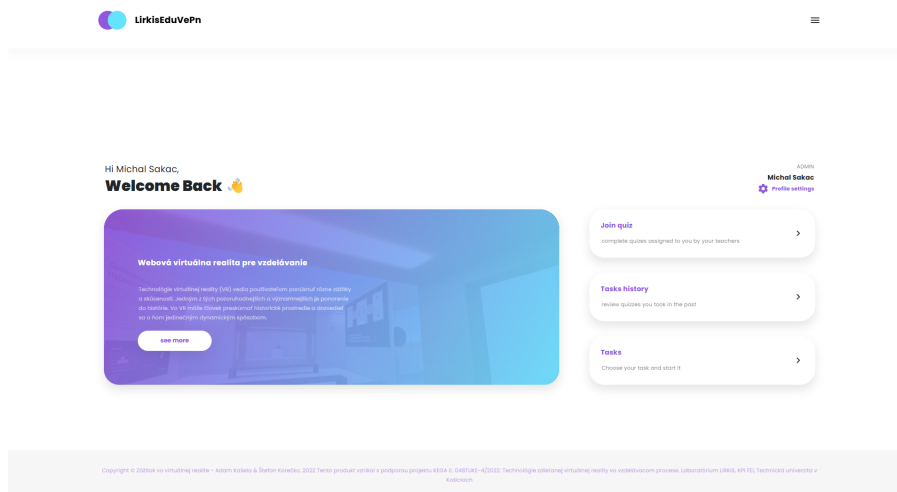
Tento program ponúka užívateľom možnosť vykonávať úlohy v virtuálnom prostredí na základe predom vytvorených scénárov. Okrem toho užívateľ môže vytvárať a upravovať vlastné scénáre a má prístup k histórii svojich predchádzajúcich pokusov. V tejto kapitole nájdete podrobný návod, ako používať aplikáciu.

B.3.1 Prihlasovacia obrazovka

Pri spustení aplikácie sa zobrazí prihlasovacia obrazovka, kde môžete zadať svoje prihlasovacie údaje, ako je používateľské meno a heslo. Ak ste ešte nevytvorili účet, máte možnosť sa zaregistrovať prostredníctvom príslušného odkazu. Po zadaní správnych prihlasovacích údajov a kliknutí na tlačidlo **Prihlásiť sa**, budete presmerovaní do hlavného rozhrania aplikácie. Na obrázku B.4 je možné vidieť prihlasovacie okno.



Obrázok B.4: Prihlasovacie okno



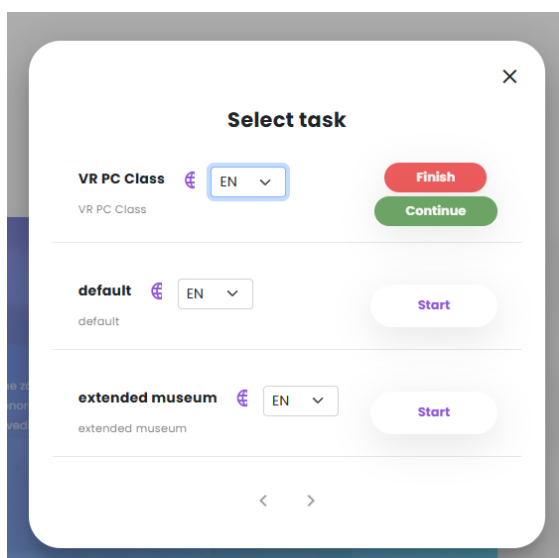
Obrázok B.5: Hlavná obrazovka

B.3.2 Hlavná obrazovka

Hlavné menu pre žiakov bolo navrhnuté tak, aby poskytovalo prehľad o všetkých úlohách, ktoré im učiteľ prideliť, a zároveň im umožňovalo zobrazíť si výsledky. Hlavnú obrazovku je možné vidieť na obrázku B.5. Pri výbere niektorej z úloh má užívateľ na výber z niekoľkých jazykov, taktiež má možnosť pokračovať v úlohe ak ju už začali alebo danú úlohu ukončiť, vidno na obrázku B.6. Okrem toho obsahuje aj možnosť prístupu k nastaveniam profilu a zmeny informácií o účte.

B.3.3 Dashboard pre učiteľov a adminov

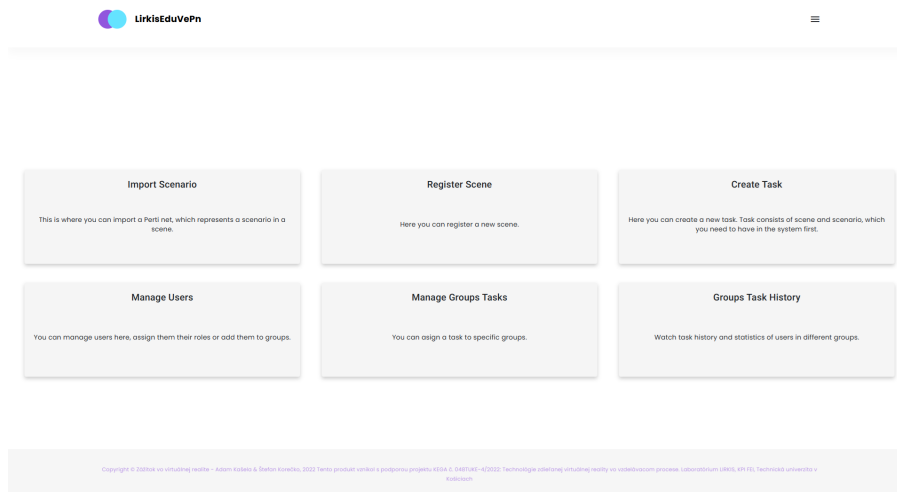
Dashboard pre učiteľov a adminov je navrhnutý tak, aby im poskytoval efektívne nástroje na správu a organizáciu vzdelávacieho obsahu a používateľov. Tento



Obrázok B.6: Výber úlohy

dashboard môžete vidieť na obrázku B.7. Nasledujúce sú niektoré z hlavných funkcií, ktoré tento dashboard ponúka:

- **Vytvoriť scénar** Učitelia a admini majú možnosť vytvoriť nový scénar, kde pridajú scénar v podobe Petriho siete a jazykový súbor s popiskami.
- **Vytvoriť scénu** Pri vytváraní scénarov učitelia a admini môžu použiť jednotlivé scény, ktoré sú už predpripravené.
- **Vytvoriť úlohu** Učitelia a admini majú možnosť vytvoriť úlohy pre študentov, kde zadajú názov, popis, scénar a scénu použitú pre úlohu.
- **Spravovať používateľov** Pre administrátorov je dostupná možnosť spravovať používateľov systému. Môžu vytvárať, upravovať a odstraňovať používateľské účty, priradiť ich do skupín, a spravovať ich oprávnenia a prístup k obsahu.
- **Spravovať skupiny** Učitelia a admini majú možnosť vytvárať a spravovať skupiny študentov. Tieto skupiny môžu slúžiť na organizáciu študentov podľa tried, kurzu alebo iných kritérií. Učitelia môžu priradiť úlohy skupinám a sledovať ich pokrok a výsledky.
- **História úloh skupín** Tento dashboard tiež umožňuje prehliadať históriu úloh skupín. Učitelia a admini majú prístup k predošlým výkonom a výsledkom študentov v rámci jednotlivých úloh. Môžu si pozrieť štatistiky a pokrok skupín v čase. Ako je možné vidieť na obrázku B.8, po vybratí skupiny má učiteľ k dispozícii všetky vykonané úlohy študentov v rámci



Obrázok B.7: Používateľský dashobard

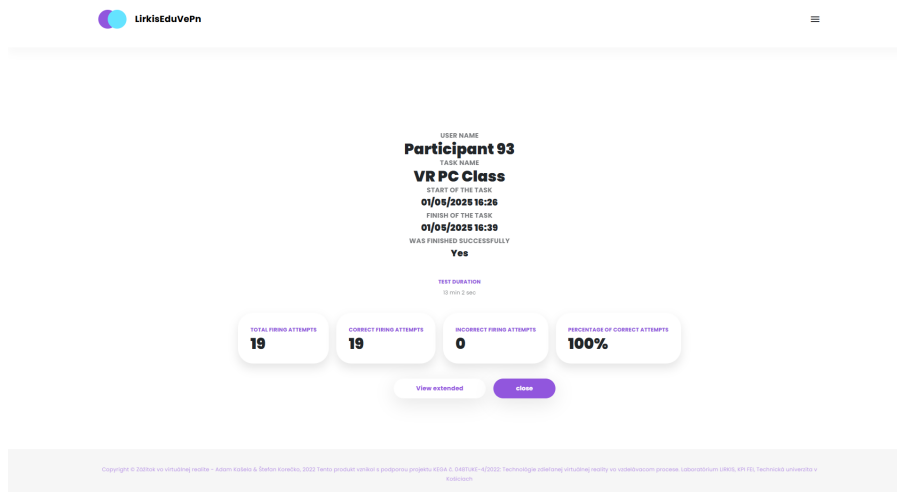


Obrázok B.8: Prehľad historie úloh pre skupinu

skupiny. Pomocou tlačidla **preview** si vie zobrazíť štatistiky pre danú úlohu, to je možné vidieť na obrázku B.9. Následne pomocou tlačidla **View extended** si môže pozrieť odpálenia jednotlivých prechodov spolu s informáciami či bol prechod odpálený úspešne, kedy sa odpálil a samozrejme aj názov prechodu.

B.3.4 Virtuálna scéna

Virtuálna scéna je prostredie vytvorené v rámci programu, ktoré umožňuje interakciu a simuláciu rôznych situácií. Je to priestor, kde sa odohrávajú úlohy a aktivity pre používateľov. Scéna obsahuje rôzne typy interakcií, ktoré majú používateľom napomôcť alebo umožniť dokončiť stanovené úlohy v scéne.



Obrázok B.9: Prehľad historie úlohy jednotlivca

Ďalej popíšeme iba interakcie z našej scény. Ak vás zaujímajú iné scény, pozrite si používateľskú príručku v predchádzajúcich prácach, [1], [2], [3] a [4].

V našej scéne máme úlohy zamerané na skladanie počítača a výmenu počítačových komponentov. Preto je hlavné interakciu sú spájanie a odpájanie počítačových komponentov. V prípade ovládania počítačom s klávesnicou a myšou, používateľ musí umiestniť zelený kurzor na objekt a následne kliknutím a držaním ľavého tlačidla počítačovej myši môže následným pohybom prenášať 3D objekt. V prípade ovládania s VR prilbou a ovládačmi, používateľ má 2 možnosti chytenia objektu pomocou ovládačov. Z jeho ľavej ruky vychádza ukazovateľ, ktorý je potrebné pohybom ruky namieriť na objekt, ktorý chce uchopiť. Následne stlačením a držaním aspoň jedného zo spodných tlačidiel na ovládači sa objekt uchopí a následným pohybom ho môže prenášať. Práva ruka je viac realistická, pretože celú ruku musí používateľ umiestniť na objekt, následne stlačením a držaním aspoň jedného zo spodných tlačidiel na ovládači sa objekt uchopí a následným pohybom ho môže prenášať. Ľavá ruka je teda určená na chytanie objektov na diaľku, alebo zo zle prístupných miest, a pravá ruka je určená na uchopenie objektov, ktoré sú blízko používateľa.

Na spojenie objektov je potrebné preniesť počítačový komponent na miesto kde patrí. Ak na danom mieste nie je prichytený iný komponent, tak sa komponent automaticky prichytí hneď ako sa dotkne miesta kde patrí. Na obrázku B.10 je vidieť grafickú kartu, ktorá je odpojená od základnej dosky. Následne na obrázku B.11 je vidieť grafickú kartu, ktorá je už pripojená k základnej doske. V prípade odpájania komponentu treba komponent chytiť a presunúť na iné miesto. Pokiaľ je komponent odpojený a naďalej držaný, tak sa nemôže automaticky



Obrázok B.10: Odpojená grafická karta

prichytiť na žiadne miesto. Preto je potrebné komponent pustiť a potom sa môže znovu prichytiť na miesto.

Ak sa komponenty pospájajú tak, ako si to scenár vyžaduje, tak sa úloha automaticky nastaví ako splnená. Splnenie úlohy je sprevádzané zvukovým znamením. Rôzne zvukové znamenia je počuť aj pri splnení všetkých úloh, ale aj pri prichytení a odpojení nejakého z komponentov. Po dokončení všetkých úloh sa používateľovi zobrazí okno s textom, ktoré mu oznámi, že všetky úlohy úspešne splnil.



Obrázok B.11: Grafická karta pripojená k základnej doske

C Systémová príručka

Systémová príručka, rovnako ako práca, priamo nadväzuje na práce Adama Kašelu [1], Dmytra Demianenka [2], Lukáša Pisarčíka [3] a Juraja Rudyho [4] a niektoré aspekty systémovej príručky, ktoré sa našou prácou nijak nemenili sme ponechali v pôvodnom stave.

C.1 Funkcia programu

Táto práca sa zameriava na systém *LirkisEduVePn*. Systém je určený na virtuálne výučbové prostredia ovládateľné pomocou počítača, VR prilby alebo mobilného zariadenia. Poskytuje učiteľom dohľad nad úlohami a ponúka administratívne funkcie. Cieľom systému je zhromažďovať údaje o používateľoch počas plnenia úloh vo virtuálnych prostrediach a poskytovať podrobné štatistiky o splnených úlohách, popri tom ako sa žiaci vzdelávajú vo virtuálnych prostrediach. Integrácia možností analýzy údajov umožňuje získať prehľad v reálnom čase a personalizovanú spätnú väzbu pre používateľov, čo zlepšuje výsledky vzdelávania.

My sme do programu pridali nové virtuálne prostredie zamerané na výučbu skladania počítačov. Jej cieľom je aby sa používatelia naučili vizuálne rozpoznávať jednotlivé počítačové komponenty a vedeli ich správne poskladať do funkčného počítača. Nezameriavame sa na to, aby sa používatelia naučili každý jeden krok skladania počítača, ale skôr chceme aby získal prehľad o tom kde by sa mali jednotlivé komponenty vložiť.

C.2 Popis programu

V tejto časti opíšeme všetky komponenty implementované a použité v tejto práci. Pre komponenty, ktoré sme implementovali my poskytneme aj podrobnejšie informácie o použití.

C.2.1 Popis riešenia

Riešenie je založené na ECS (Entity-Component-System) architektúre. ECS umožňuje oddelenie dátových entít (entity) od ich správania (component) a systémov, ktoré tieto entity spracúvajú. ECS architektúru sme obohatili o webový rámec Angular, ktorý poskytuje robustné nástroje na tvorbu užívateľského rozhrania. Angular nám umožňuje efektívne spravovať komponenty a moduly, čo je veľmi dôležité pri rozsiahlejších projektoch.

Klientská časť aplikácie, implementovaná v Angulari, a aj serverová časť, implementovaná v Java, boli implementované v prácach [2], [3] a [4]. Štruktúru a popis týchto častí aplikácie nájdete v spomenutých prácach.

C.2.2 Štruktúra prostredia pre skladanie počítačov

Na výpise C.1 je zobrazená štruktúra priečinkov a súborov, ktoré sme implementovali alebo použili v našej práci na vytvorenie prostredia pre skladanie počítačov.

Výpis C.1: Štruktúra súborov použitých na vytvorenie prostredia pre skladanie počítačov

```
LirkisEduVePn/LirkisEduVePn-UI/src/app/components/tasks
├── js
│   ├── components
│   │   ├── attachmentBox.component.js
│   │   ├── blinkControl.component.js
│   │   ├── computerAssemblyCollisionGroup.component.js
│   │   ├── crouch.component.js
│   │   ├── mobileViewHandler.component.js
│   │   ├── petriNetSim.component.js
│   │   ├── phaseShiftHands.component.js
│   │   ├── pn-transition-handler.component.js
│   │   ├── sceneLanguage.component.js
│   │   ├── transitions.ts
│   │   ├── visibilityBasedOnPetriNet.component.js
│   │   └── vrDetectionForRaycasting.component.js
│   └── modules
│       ├── petriNet.mjs
│       ├── petriNetLoader.mjs
│       └── userActivityLogger.js
└── scenes
```

```

└─ computer_assembly
    │─ computer_assembly.component.css
    │─ computer_assembly.component.html
    │─ computer_assembly.component.spec.ts
    │─ computer_assembly.component.ts
    └─ scripts
        └─ sceneScript.ts

```

C.3 Popis implementovaných súborov

V tejto práci sme implementovali iba A-Frame komponenty pre našu scénu. Všetky ostatné komponenty boli implementované v predošlých prácach [1], [2], [3] a [4]. Štruktúru súborov a priečinkov, ktoré sme implementovali alebo použili v našej práci na vytvorenie prostredia pre skladanie počítačov, nájdete v podsekcii C.2.2. Naše nové implementované A-Frame komponenty sú:

- **attachmentBox.component.js** - tento komponent rieši problematiku spájania a oddeľovania 3D objektov.
- **computerAssemblyCollisionGroup.component.js** - tento komponent rieši problematiku kolízií medzi 3D objektmi.
- **crouch.component.js** - tento komponent umožňuje zníženie kamery hráča v scéne.
- **visibilityBasedOnPetriNet.component.js** - tento komponent umožňuje zmenu viditeľnosti 3D objektov na základe scenára Petriho siete.

Komponenty, ktoré boli implementované v predošlých prácach a sú použité v našej scéne sú:

- **blinkControl.component.js** - umožňuje teleportáciu, ako alternatívny spôsob pohybu, pomocou VR ovládača.
- **mobileViewHandler.component.js** - rieši problematiku ovládania scény pomocou mobilného zariadenia.
- **petriNetSim.component.js** - najdôležitejší komponent aplikácie, zodpovedný za simuláciu a riadenie scény pomocou Petriho siete.
- **phaseShiftHands.component.js** - pomocný komponent pre super-hands slúži na to, aby sme virtuálnymi rukami nenarážali do objektov v scéne.

- **pn-transition-handler.component.js** - počúva na správy udalostí a podľa nich posiela pokusy o odpálenie prechodov v Petriho sieti.
- **sceneLanguage.component.js** - umožňuje zobrazovať informácie v scéne v jazyku, ktorý si užívateľ zvolí.
- **transitions.ts** - obsahuje rozhrania pre prechody a miesta v Petriho sieti.
- **vrDetectionForRaycasting.component.js** - slúži ako detekciu prepnutia do VR režimu. Následne po presune do VR režimu sa zo scény odstráni raycaster, ktorý slúži pre ovládanie počítačom.

Mimo komponentov naša scéna používa aj niektoré moduly, ktoré sú implementované v predošlých prácach. Tieto moduly sú:

- **petriNet.mjs** - implementuje štruktúru a logiku pre Petriho sieť.
- **petriNetLoader.mjs** - pre načítanie a inicializáciu Petriho siete zo súboru vo formáte cpn.
- **userActivityLogger.js** - obsahuje implementáciu skriptu pre zaznamenávanie aktivity užívateľa počas vykonávania úlohy.

Okrem spomenutých komponentov a modulov sme pridali aj scénu pre naše nové prostredie. Súbory tejto scény sú:

- **computer_assembly.component.css** - CSS štýly pre zobrazenie ovládacích prvkov pre mobilné zariadenia.
- **computer_assembly.component.html** - obsahuje A-Frame scénu, v ktorej definujeme všetky assety, objekty vytvorené z assetov a komponenty, ktoré sa na objektoch používajú.
- **computer_assembly.component.spec.ts** - testy pre komponent scény.
- **computer_assembly.component.ts** - obsahuje skript pre načítanie a inicializáciu scény, s tým je spojené aj získanie scenára a jazykového súboru pre túto scénu.
- **sceneScript.ts** - obsahuje definície prechodov a miest v Petriho sieti, niektoré prechody majú logiku navyše, ako napríklad zobrazenie indikátora dokončenej úlohy, alebo prehratia zvuku.

C.4 Popis implementovaných A-Frame komponentov

attachmentBox komponent

Tento komponent je zameraný na spájanie a oddeľovanie 3D objektov. V našom prípade sa jedná o spájanie a oddeľovanie počítačových komponentov.

Tento komponent sme v našej scéne pridávali na neviditeľné 3D boxy, ktoré boli vložené do iných 3D modelov. Ak sa vybraný počítačový komponent dotkol boxu, tak sa prichytil. Prichytenie znamená, že sa jeho rodičovským objektom stal rodičovský objekt boxa, pričom sa mu upraví pozícia a rotácia na pozíciu a rotáciu boxa. Box prestane robiť kolízie a prichytený objekt sa nastaví na kinematický.

Ak je prichytený objekt uchopený hráčom, tak sa znovu stane dynamickým a jeho rodičovským objektom sa stane samotná scéna. Keď hráč pustí tento objekt, tak tento neviditeľný box začne znovu robiť kolízie, to znamená, že sa znovu môže nejaký objekt prichytiť.

Pri prichytení a oddelení objektu sa posielajú správy udalostí, *attached*, *play-attached-sound*, *dettached*, *play-dettached-sound*.

Parametre tohto komponentu sú:

- **attachmentGroup** - uvádza sa tu jedna zo skupín z komponentu *computerAssemblyCollisionGroup.component.js*, rozhoduje o tom aké objekty môžu mať s týmto neviditeľným boxom kolízie.
- **allowedIDs** - je zoznam identifikátorov objektov, ktoré môžu mať kolízie s týmto boxom, kontroluje sa pri kolízii boxu a objektu, prázdny zoznam znamená, že sa môže prichytiť akýkoľvek objekt.
- **petriNetPlaces** - je zoznam miest Petriho siete, ktoré sa pri načítaní scény skontrolujú a podľa tokenov v týchto miestach sa vie objekt automaticky prichytiť, každý záznam musí byť spárovaný so záznamom v *preattached* parametri.
- **preattached** - je zoznam identifikátorov objektov, ktoré sa majú prichytiť pri načítaní scény, každý záznam musí byť spárovaný so záznamom v *petriNetPlaces* parametri.

computerAssemblyCollisionGroup komponent

Tento komponent je zameraný na definovanie kolíznych skupín a masiek pre 3D objekty. Kolízna skupina označuje o aký typ počítačového komponentu sa jed-

ná, pričom kolízna maska určuje s ktorými kolíznymi skupinami môže mať daný objekt kolízie.

V našej scéne tento komponent používame aj na priradzovanie komponentov z fyzikálnej knižnice *Ammo.js*. Hlavne preto, že objekty ktoré majú GTLF 3D model, čo sú všetky naše počítačové komponenty, sa najprv musia načítať a až potom sa im môžu priradiť fyzikálne vlastnosti. Tento komponent taktiež zabezpečuje, že sa fyzikálne vlastnosti priradia v správnom poradí.

Parametre tohto komponentu sú:

- **group** - názov skupiny, podľa ktorej sa nastaví kolízna skupina komponentu fyzikálnej knižnice.
- **manualFit** - označuje či je potrebné manuálne nastaviť rozmery kolízneho boxu.
- **halfExtents** - možnosť manuálneho nastavenia rozmerov kolízneho boxu pomocou 3 rozmerov.
- **heightfieldDistance** - možnosť manuálneho nastavenia rozmerov kolízneho boxu, musí byť uvedený spolu s *heightfieldData* parametrom, označuje vzdialenosť jednotlivých bodov mriežky kolízneho boxu.
- **heightfieldData** - možnosť manuálneho nastavenia rozmerov kolízneho boxu, musí byť uvedený spolu s *heightfieldDistance* parametrom, označuje výšky v jednotlivých bodoch mriežky kolízneho boxu.
- **initializeAfter** - možnosť upraviť poradie inicializácie fyzikálnych vlastností na objektoch, ak je uvedený identifikátor iného objektu, tak sa fyzikálne vlastnosti nastaví až po inicializácii tohto objektu.

crouch komponent

Tento komponent je zameraný na zníženie kamery hráča v scéne. Predpokladá sa, že je tento komponent priradený na objekt kamery hráča, ktorá je vnorená do objektu hráča.

Parametre tohto komponentu sú:

- **keyCode** - kód klávesy, ktorá sa má stlačiť na zníženie kamery, predvolená hodnota je klávesa C, nepoužívajte *Ctrl* klávesu, lebo klávesová skratka *Ctrl* + *W* zatvára okno prehliadača.

- **toggle** - označuje či sa má po pustení klávesy kamera zvýšiť, alebo či sa stlačením klávesy iba prepína medzi zníženou a normálnou výškou.
- **crouchHeight** - výška kamery pri znížení,
- **standHeight** - výška kamery pri normálnej výške.
- **transitionSpeed** - rýchlosť prechodu medzi výškami kamery.

visibilityBasedOnPetriNet komponent

Tento komponent je zameraný na zmenu viditeľnosti 3D objektov na základe scenára Petriho siete. V Petriho sieti uvediete oddelené miesto. Ak v ňom nebude žiadny token alebo miesto sa v Petriho sieti nenachádza, tak sa objekt skryje, inak ostáva viditeľný.

Parametre tohto komponentu sú:

- **place** - názov miesta v Petriho sieti.

C.5 Využité technológie

Technológie, ktoré boli použité na vytvorenie aplikácie. V zozname je uvedená aj minimálna verzia technológie, ktorá je potrebná na úspešné spustenie programu:

- **A-Frame** (1.4.1)
- **Angular** (15.0.4)
- **Bootstrap** (5.2.3)
- **Java** (17)
- **PostgreSQL** (15.3)
- **Docker**

C.6 Spustenie programu

Na lokálny vývoj a spustenie aplikácie lokálne je viacero spôsobov. Počas našej práce sme využívali iba Docker. Vďaka tomu sme mali istotu, že sa naše prostredie bude správať rovnako na našich počítačoch, ako aj na serveri. Ak vás zaujíma spôsob ako jednotlivé časti aplikácie spustiť lokálne bez Dockeru, tak si pozrite systémové príručky z predošlých prác [2], [3] a [4].

V priečinku projektu sa nachádza súbor *docker-compose.yml*. Ten obsahuje všetky potrebné nastavenia na spustenie aplikácie. Pre ovládanie servisov cez tento súbor používame príkaz „docker compose [parametre]“ pri vývoji cez Windows a „docker-compose [parametre]“ pri vývoji cez Linux. Na spustenie aplikácie použijeme parameter „up“. Aby aplikácia ostala spustená aj po zatvorení terminálu, použijeme parameter „-d“ (detached). Ak sme urobili nejaké zmeny v kóde a chceme ich aplikovať, použijeme parameter „-build“ na prekompilovanie aplikácie. Teda počas celého vývoja používame príkaz, ale aj pri prvom spustení nám stačí príkaz „docker compose up -d -build“. Ak chceme aplikáciu zastaviť, použijeme príkaz „docker compose down“. Pre pridanie administrátorského účtu môžete použiť pgAdmin kontajner s dotazmi z výpisu B.3, ktoré sme popísali v používateľskej príručke v podsekcii B.2.2.

C.7 Nasadenie na server

Pre nasadenie novej verzie na server je potrebný nasledovný postup:

1. **Nahrať potrebné obrazy kontajnerov do dockerhubu** - Pre nahratie obrazov(container image) do dockerhubu odporúčame používať tagy. Tie slúžia na rozoznávanie rôznych verzií nahratých obrazov. Najskôr si vylistujeme zoznam obrazov pomocou príkazu „docker image ls“. Potom si nad konkrétnym obrazom vytvoríme tagovanú kópiu pomocou príkazu „docker tag [ID] [dockerhub account]/[image name]:[tag]“. Následne pushneme obraz do dockerhubu pomocou príkazu „docker push [dockerhub account]/[image name]:[tag]“.
2. **Stiahnutie obrazov na server** - Po prihlásení sa na server je potrebné obrazy z dockerhubu stiahnuť. Na to použijeme príkaz „docker pull [dockerhub account]/[image name]:[tag]“. Keď máme stiahnuté obrazy, potrebujeme si buď stiahnuť, alebo vytvoriť súbor *docker-compose.yml*.
3. **Spustenie systému** - Pre spustenie systému použijeme príkaz „docker compose up -d -build“. V tomto prípade nám prepínač „-build“ nainštaluje nové obrazy a prepínač „-d“ nám uvoľní terminál po spustení kontajnerov, v opačnom prípade bude terminál blokováný do doby, kým sa kontajnery zase nevypnú.