

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informatiky a kvantitativních metod

Virtuální realita v Unity3D

Bakalářská práce

Autor: Tomáš Bareš
Studijní obor: Aplikovaná informatika

Vedoucí práce: Ing. Bruno Ježek, Ph.D.

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 16.8.2019

vlastnoruční podpis

Jméno a Příjmení

Poděkování:

Tímto bych chtěl poděkovat vedoucímu mé bakalářské práce panu Ing. Brunovi Ježkovi, Ph.D. za metodické vedení práce, velkou trpělivost a cenné rady týkající se této práce. Také bych chtěl poděkovat FIM UHK za zapůjčení zařízení pro virtuální realitu HTC Vive, na které jsem mohl testovat svá řešení. A nakonec chci poděkovat i rodině, přátelům a kolegům z práce za podporu a pomoc s testováním.

Anotace

Tématem bakalářské práce je Virtuální realita v Unity3D. Práce se v teoretické části zabývá konceptem virtuální reality a poskytuje informace o vzniku myšlenky, principech, negativních účincích a oblastech, ve kterých je často využívána. Dále je představen herní engine Unity3D jako jeden z možných softwarových řešení podporujících virtuální realitu. Je popsána základní struktura programu, poskytované služby a dostupné platformy umožňující sestavení výsledné aplikace. V praktické části je navržen konkrétní systém využívající popsané principy a technologie. Následovně je rozebrán vlastní postup implementace při řešení projektu. Implementace se zabývá samotnou tvorbou aplikace, kde se blíže věnuje řešení interakcí a s nimi spojenými pohyby po scéně a zprovozněním automatické navigace. V poslední části jsou shrnuty výsledky společně s řešenými problémy, které při tvorbě aplikace nastaly. Na závěr jsou zhodnoceny výsledky a vlastní přístupy k řešení.

Annotation

Title: Virtual reality in Unity3D

The topic of this thesis is Virtual Reality in Unity3D. The theoretical part deals with the concept of virtual reality and provides information on the origin of the idea, principles, negative effects and areas in which it is often used. The Unity3D game engine is introduced as one of the possible software solutions supporting virtual reality. This thesis describes the basic structure of the program, the services provided and the available platforms to build the resulting application. The practical part describes the design of a concrete system using the described principles and technologies. Subsequently, the actual implementation procedure of the project is analyzed. The implementation deals with the creation of the application itself. Furthermore, it is devoted more specifically to the solution of interactions and their associated movement around the scene and the initiation of automatic navigation. The last part summarizes the results together with the problems that occurred during the creation of the application. Finally, the results and personal approaches to the solution are evaluated.

Obsah

1	Úvod.....	1
2	Virtuální realita.....	2
2.1	Historie.....	2
2.2	Současnost	3
2.3	Využití systémů VR.....	4
2.4	Head-mounted display	4
2.5	Negativní účinky VR na člověka.....	5
3	Herní engine Unity3D	6
3.1	Typy licencí systému Unity3D	6
3.2	Podporované platformy	7
3.3	Programovací jazyky.....	7
3.3.1	Boo.....	8
3.3.2	UnityScript	8
3.3.3	C#.....	9
3.4	Asset	9
3.5	Panely.....	9
3.6	Komponenty.....	10
3.7	Služby a vnitřní systémy.....	12
3.7.1	Unity Collaborate.....	12
3.7.2	Unity In-App Purchase.....	13
3.7.3	Unity Ads.....	13
3.7.4	Unity Asset Store.....	13
4	Definice cílů a metodika zpracování.....	15
4.1	Cíl systému.....	15

4.2	Výběr technologií	15
5	Zdroje dat do scény.....	18
5.1	API.....	18
5.2	JSON.....	18
5.3	IS/STAG.....	19
5.3.1	REST rozhraní.....	19
6	Návrh a implementace systému	22
6.1	Vytvoření projektu.....	22
6.2	Úprava scény	22
6.3	Hierarchické dělení tříd.....	23
6.4	Pohyb ve scéně	24
6.4.1	Teleport skokem v daném směru.....	24
6.4.2	Teleport na danou pozici.....	25
6.4.3	Posun daným směrem	25
6.4.4	Automatická navigace.....	25
6.5	Interakce.....	27
6.6	Integrace externích dat do scény	28
7	Výsledky.....	30
8	Závěry a doporučení	34
	Seznam použité literatury	36

Seznam obrázků

Obrázek 1: Hlavní podporování platformy (Daniel Bratcher 2013).....	7
Obrázek 2: Procentuální využití jazyků uživateli (Aleksandr 2014).....	8
Obrázek 3: Objekt s přiřazenými komponentami (Jeremiah van Oosten 2012).....	11
Obrázek 4: Unity Asset Store (Nathan Ventura 2017)	14
Obrázek 5: Headset HTC Vive (HTC Corporation 2019).....	17
Obrázek 6: Porovnání JSON a XML (Hoi 2017)	19
Obrázek 7: Příklad volání metody s parametry v prostředí Postman.....	21
Obrázek 8: Model budovy FIM v Unity3D	23
Obrázek 9: Příklad vygenerovaného NavMesh objektu.....	26
Obrázek 10: Učebna se svým rozvrhem a UI pro navigace.....	27
Obrázek 11: Příklad interakcí ve VR prostřednictvím laserových ukazovátek	28
Obrázek 12: Příklad vygenerovaného rozvrhu ze získaných dat	29
Obrázek 13: Příklad objektu Rig ve scéně SteamVR Home	31
Obrázek 14: Příklad navigační linky usnadňující orientaci.....	32
Obrázek 15: Vytvořené rozvrhy u učeben z dat za školní API.....	32
Obrázek 16: GUI pro desktopovou verzi s možností volby zobrazovaných dat	33

Seznam tabulek

Tabulka 1: Specifikace HTC Vive (VIVE™ 2019).....	16
---	----

1 Úvod

Virtuální realita je fenoménem dnešní doby. Objevuje se ve stále více oblastech našeho života. Nachází uplatnění na různých místech lidské činnosti, ať už se jedná o vojenský, strojírenský nebo stavební průmysl. V posledních letech se technologie používaná pro virtuální realitu posunula bleskově dopředu a stala se mnohem více dostupnější. Napomohl tomu její posun na poli vývoje herních aplikací, díky stále zvětšující se podpoře u nástrojů umožňujících vývoj aplikací pro tuto technologii. V současnosti jedním z nejoblíbenějších softwarových nástrojů je Unity3D, které je díky jeho dobré podpoře technologií virtuální reality programátory často využíváno.

Hlavním cílem této práce v teoretické části je seznámení čtenáře se základními pojmy, jako je virtuální realita, herní engine nebo API, které jsou pro tuto práci základními stavebními kameny. Teoretická část se z počátku práce věnuje představení principů virtuální reality, její historii, oblastí využití a jejím případným negativním účinkům, které může mít na jejího uživatele. Další kapitola se věnuje popisu softwarového nástroje Unity3D, jeho struktuře, službám, které poskytuje a platformám, pro které umožňuje sestavení aplikace. V poslední teoretické kapitole jsou vysvětleny zkratky API a IS/STAG a s nimi spojené technologie.

Získané znalosti jsou využity při návrhu systému a jeho tvorbě v praktické části. Praktická část má za cíl realizovat aplikaci spojením zmíněných technologií pro zobrazení herní scény. Scéna aplikace bude zasazena do prostředí s budovou Fakulty informatiky a managementu, ve které bude umožněn pohyb, interakce a budou zaintegroována data z externího zdroje.

Vlastní návrh výsledného systému je popsán v kapitole Návrh a implementace systému. A dosažené výsledky jsou shrnuty v kapitolách Výsledky a Závěry a doporučení.

2 Virtuální realita

Virtuální realita (dále VR) je koncept umožňující simulovat skutečný svět a je spojována s technologiemi a zařízeními, která poskytují člověku možnost zobrazit informace modelovaného světa a provádět s nimi interakce. Primárním cílem této technologie je vytvoření co nejuvěrohodnější simulace reálného světa. Simulace je zobrazována člověku prostřednictvím zařízení, které umí jejímu uživateli ovlivnit smysly tak, že se cítí jako uprostřed zobrazované scény. Proto se namísto používání klávesnice, myši a monitoru využívají speciální brýle, ovladače a sluchátka. Což představuje obrovský skok ve způsobu interakce s počítačem a vizualizací informací (Jun 2017).

Technologie spojené s VR se vyvíjí dlouhou dobu, ale mezi běžné uživatele se ve větším měřítku začala dostávat teprve nedávno. Vděčíme za to rychlému pokroku vědy v této oblasti a firmám zabývajících se touto technologií. V průběhu času a postupným vývojem se začala přesouvat od armádních simulačních zařízení i k herním konzolím. Současné systémy umožňují prostřednictvím prostředků VR poskytnout lepší požitek díky ovlivňování našich smyslů jako je zrak, sluch a hmat. Poskytnuté vjemy mají člověka pohltnout natolik, že přestává vnímat okolí reálného světa, a plně se vžívá do role ve virtuálním světě.

2.1 Historie

Za počátky virtuální reality lze považovat malbu z 19. století znázorňující s 360° panoramatickým pohledem bitvu u Borodino. Jejím smyslem, stejně jako smyslem virtuální reality je zaplnit zorné pole diváka, tak aby se cítil uprostřed scény, na kterou se dívá. V České republice je podobná scéna znázorňující Bitvu u Lipan vystavena na Pražském výstavišti (Virtual Reality Society 2017).

V roce 1929 vznikl stroj s názvem „Linkův trenážér“ pojmenovaný po svém tvůrci Edwardu Linkovi (Virtual Reality Society 2017). Pravděpodobně se tehdy jednalo o první příklad komerčního leteckého simulátoru. Tento motorem poháněný stroj dokázal nasimulovat letecké turbulence či poruchy letadla. Za druhé světové války o tento výrobek projevil zájem americká armáda za účelem školení svých pilotů (Virtual Reality Society 2017).

Termín virtuální realita se objevil teprve v roce 1987, kdy Jaron Lanier s Tomem Zimmermanem uvedli na trh první komerční příslušenství pro VR tvořené zobrazovacími brýlemi, rukavicemi a sluchátky (Virtual Reality Society 2017; Lowood 2018).

Dalším z důležitých mezníků ve vývoji a nahlížení na virtuální realitu byl první díl sci-fi filmu Matrix vydaný roku 1999, kde hlavní postavy bojují za přežití svého reálného světa v tom virtuálním, ve kterém se na ně následky jako city, zranění a bolest přenáší zpět skrze sondu zavedenou do mozku (Jun 2017).

2.2 Současnost

Virtuální realita je v současné době oproti minulosti mnohem více rozšířená mezi běžné uživatele. Velmi tomu přispěl její vstup do herního průmyslu. Řada uživatelů počítačových her, které podporují technologii VR, si pořídilo zařízení do svých domácností, a to i přes to, že ceny jsou ještě stále vysoké a v době psaní této práce se pohybují v řádech tisíců až desetitisíců korun v závislosti na značce a kupovaném modelu.

Nejznámějšími společnostmi prodávající komplexnější zařízení pro virtuální realitu jsou HTC, PlayStation, Samsung a Microsoft. Tito výrobci u svých zařízení užívají pokročilé metody snímání pohybu, interakce uživatele a moderní technologie pro realistické zobrazování obrazu vykreslované scény v reálném čase.

Na trhu existuje široká škála zařízení, která umožní zprovoznit VR jen za pomoci chytrého mobilu. U dostatečně výkonných smartphonů, si stačí pořídít brýle, do kterých se mobilní zařízení vloží a k vyzkoušení už je nutné pouze spustit aplikaci podporující VR. Tyto virtuální brýle pak bývají značně levnější oproti komplexnějším zařízením, a to díky využití komponent z mobilního zařízení jako je akcelerometr, gyroskop a magnetický senzor, které slouží k získání informací o aktuálním natočení zařízení. Technologická úroveň použitého smartphonu se následně odráží ve výsledném prožitku a dosažené kvalitě obrazu vykreslované scény. Většina takovýchto zařízení neumožňuje přímou interakci uživatele s virtuálním prostředím.

2.3 Využití systémů VR

Systémy VR mají velice široké možnosti využití, a kromě zábavního průmyslu se tato technologie využívá i v řadě dalších odvětvích. Jednou z nejrozsáhlejších oblastí je oblast vojenských aplikací jako jsou simulátory pro ovládání vojenských prostředků, řízení bezpilotních dronů či robotů a simulace strategických situací. S tím je spojené i nasazení ve zdravotnictví, kde se systémy používají pro rehabilitační prostředky pomáhající proti různým druhům ochrnutí. Užívají se také jako terapeutické pomůcky pacientů s psychickými problémy nebo tréninkové simulace operací pro chirurgy.

Virtuální realita je rozšířena také ve strojírenském a automobilovém průmyslu nebo architektuře, kde umožňuje prohlídku ještě fyzicky neexistujících výrobků či objektů. Náhled je také využit ve vzdělávání, kdy se do scén rekonstruuji těžko dostupná místa, historické události nebo zaniklé tvorové. V chemii pak může usnadnit práci ve virtuální laboratoři pro bezpečné simulace různých pokusů. Nebo podpořit kreativitu formou 3D malování.

2.4 Head-mounted display

Důležitou částí systémů VR je zprostředkování zrakového vjemu. Vizualní obraz vytváří v systémech VR hlavový displej tzv. headset nebo také head-mounted display (HMD). Obecně se jedná o zařízení poskytující uživateli obrazový, často stereoskopický, náhled do světa modelovaného ve VR (Skala 2017).

Stereoskopický headset se skládá ze dvou displejů a čoček, každou pro jedno oko, a vytváří tak uživateli stereoskopický obraz prostorové scény. Tuto myšlenku vynalezl Sir Charles Wheatstone v roce 1832 a její princip spočíval v prohlížení kreseb skrze dvě samostatná okýnka, do nichž se odrážel obraz z dvou zrcadel v úhlu 45°. Lidský mozek s pozorovaných kreseb vytvářel třírozměrný vjem (Zoe Clayton 2013).

V nynější době díky rychlým grafickým kartám headsety umožňují zobrazovat snímky v reálném čase v dostatečné rychlosti, aby vytvořil věrohodný nepřerušovaný obraz. Grafická karta zpracuje obraz zobrazovaného virtuálního prostředí do dvou snímků s posunem pro každé oko a odešle je do headsetu. Tam se snímky zobrazí pro patřičné oko pro zajištění prostorového vjemu. S tímto

způsobem zobrazování je také spojena spousta problémů týkajících se příliš velké odezvy, malého zorného pole, nízkého počtu zobrazených snímků za sekundu nebo špatné synchronizace snímků zpracovaných grafickou kartou a maximální rychlosti zobrazovaných v headsetu. Všechny z těchto problémů negativně ovlivňují výsledný prožitek z virtuální reality a mohou nepříjemně účinkovat na osobu užívající VR, což je více rozebráno v další části.

Pro lokalizaci v prostoru, úhlu naklonění a směru pohledu bývají současně s headsetem dodávány základní stanice, které HMD a ovladače snímají a umožňují získání jejich pozice a rotace ve výsledném virtuálním prostředí. Ovladače jsou pak zařízení disponující několika tlačítky nebo dotykovými snímači umožňující interakci v prostoru scény. Bohužel nemusí být vždy součástí balení při zakoupení virtuální reality.

2.5 Negativní účinky VR na člověka

Vliv virtuální reality na člověka nemusí být vždy jen pozitivní. Obzvláště pokud pohyb okolní scény zobrazený v HMD nastává bez skutečného pohybu uživatele, který zůstává stále pevně na stejném místě. Tento jev se nazývá virtual reality sickness, v angličtině znám také pod pojmem cybersickness, se kterým sdílí i některé symptomy. Tento jev může často způsobovat tzv. okulomotorické příznaky se kterými je spojeno nepohodlí, únava a dezorientovanost a v některých případech i nevolnosti žaludku nebo bolesti a motání hlavy. Také zpoždění může v této situaci hrát důležitou roli. U méně výkonných zařízeních, která nejsou schopna reagovat v dostatečné rychlosti, může nastat smyslový konflikt při pohybu či otočení hlavy, způsobený pomalým zobrazováním výsledné projekce (Jun 2017).

3 Herní engine Unity3D

Unity3D je komplexní integrované vývojové prostředí (IDE) a multiplatformní herní engine vytvořený společností Unity Technologies. První verze byla vydána v roce 2005 a od této verze se Unity rozvíjelo o řadu nových užitečných prvků. Unity3D je navrženo jako systém pro tvorbu her či aplikací pro různé typy hardwarových zařízení a podporuje vývoj pro širokou škálu softwarových platform (Balcárek 2015).

Jádro systému je Unity3D engine, které poskytuje nejzákladnější funkce jako je především vykreslování 2D nebo 3D scény, výpočty fyziky, detekce kolizí a jejich zpětná odezva, zvuk, animace, umělá inteligence a mnoho dalších.

Dobrým příkladem je hra DOOM od studia Id Software. Tato hra měla jako jedna z prvních oddělenou vlastní náplň hry (pravidla, prostředí, zvuky) od samotného jádra hry (detekce kolizí, fyzické výpočty, vykreslování grafiky). Toto řešení se ukázalo jako velmi efektivní a využitím tohoto přístupu mohlo studio dále licencovat jádro ostatním firmám. Těm pouze stačilo vytvořit vlastní prostředí a pravidla hry a nemusely se zabývat nízko-úrovňovými aspekty hry (Kovářík 2017).

Unity také disponuje rozsáhlou komunitou a fórem, kde uživatelé spolu mohou sdílet své poznatky a vytvořené projekty pro inspiraci ostatních. Kromě fóra Unity nabízí službu s názvem Answers, ve které si uživatelé odpovídají na dotazy, jež se týkají jejich projektů (Dansie 2013).

Prostředí Unity editoru je tvořeno z několika panelů, které umožňují různé funkcionality. Tyto panely ve většině případů pracují s prvky GameObject, což jsou základní objekty, fungující jako pouzdra nebo složky pro komponenty, které definují objektům ve scéně požadované funkce a vlastnosti. Každý GameObject od vytvoření vlastní komponentu Transform, která je blíže popsána v sekci Komponenty. V této práci jsou popsány pouze nejčastěji využívané panely a komponenty, které byly nejvíce využívány za účelem tvorby této bakalářské práce.

3.1 Typy licencí systému Unity3D

Unity3D je dostupné ve 3 verzích: Personal, Plus a Pro. Personal verze obsahuje všechny hlavní funkce enginu a je zdarma dostupná všem, kteří

při komerčním užití nepřekročí zisk 100000 \$ za rok. Při překročení je nutné zakoupit verzi Plus, která oproti Personal disponuje několika výhodami jako 20% slevy do Asset Store na některé produkty, tvorbu vlastní načítací obrazovky, zpětné vazby při pádu aplikace a možnosti změny designu editoru. Poslední verze Pro obsahuje všechny prvky jako obě předešlé verze, a navíc umožní přístup do zdrojového kódu, prémiovou podporu teamu Unity a tričko z limitované edice Unity Pro. (Unity Technologies 2019)

3.2 Podporované platformy

Jednu z největších výhod, které Unity3D má, je jeho široká podpora platform, na které uživatel může hru sestavit. Mezi možnosti patří vydání aplikace pro mobilní zařízení, webové prohlížeče, konzole nebo pro desktopová zařízení. Platformy podporované pro stolní počítače a notebooky jsou Windows, Mac, Linux nebo Windows Store. U mobilních zařízení je podpora pro iOS, Android, Windows Phone a Tizen. Mezi podporované herní konzole patří PlayStation, Xbox, PlayStation Vita a Wii U, a aplikace pro webové prohlížeče se spouští přes WebGL rozhraní (Varga 2016).



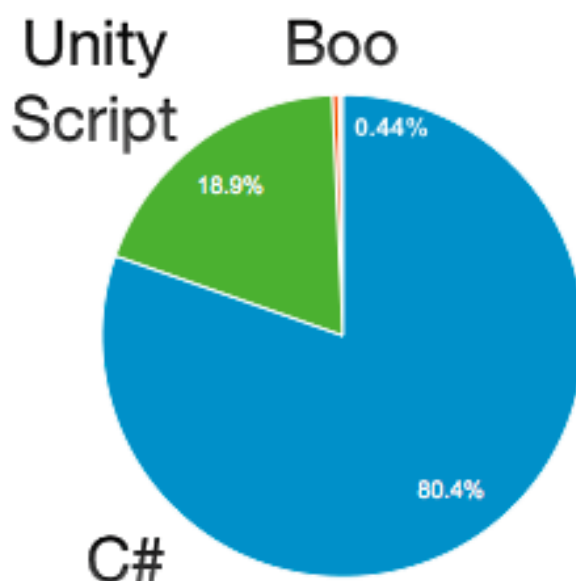
Obrázek 1: Hlavní podporované platformy (Daniel Bratcher 2013)

3.3 Programovací jazyky

V systému Unity lze programovat v jazycích C#, UnityScriptu a v Boo, ve kterém již není nadále možnost vytvářet nové skripty. V posledních letech je nejvíce rozšířený pouze jazyk C# na rozdíl od ostatních enginů, které mají jako hlavní nebo jediný programovací jazyk C++ (Unity 2019)

Při volbě jazyka je dobré zohlednit předchozí zkušenosti a znalosti s programovacími jazyky. V případě, že znalosti programátora spadají spíše do objektově orientovaných programovacích jazyků, jako je například Java, je dobré

vyvíjet v jazyku C#. Naopak pokud vývojář nemá dostatek znalostí o objektovém přístupu, je vhodné zvolit UnityScript. Ale jelikož Unity z dlouhodobého hlediska chce podporovat jen C# a veškeré skripty migrovat z UnityScriptu do C#, vyplatí se vybrat tento jazyk. K roku 2014 Unity3D uvádí graf (Obrázek 2) s poměry využívaných programovacích jazyků. Již v té době byl nejvíce využívaným jazykem právě C#, ve kterém programovalo 80,4 % uživatelů, nyní může být procento zastoupené vývojáři v jazyce C# ještě vyšší (Varga 2016).



Obrázek 2: Procentuální využití jazyků uživateli (Aleksandr 2014)

3.3.1 Boo

Boo se pro skriptování v Unity používalo jenom zřídka, jak je vidět na obrázku. Od verze 5 byla odstraněna možnost vytvořit skript v tomto jazyce. Stále je možnost vytvořené skripty spustit, ale tvoření nových skriptů je umožněno jen pro C# a UnityScript (Varga 2016).

3.3.2 UnityScript

UnityScript je jazyk používaný v Unity pro tvoření skriptů, který je velmi podobný JavaScriptu, což je interpretovaný, multiplatformní, objektově orientovaný skriptovací jazyk. Ačkoli UnityScript a JavaScript vypadají syntakticky velmi

podobně, jedná se o velmi odlišné programovací jazyky s rozdílnou sémantikou (Unify Community Wiki 2014). UnityScript byl použit v jedné z předchozích verzí, s cílem přilákat řady web developerů, kteří mají zájem o herní vývoj ve 3D (Kovářík 2017).

3.3.3 C#

C# je objektově orientovaný programovací jazyk vyvinutý firmou Microsoft, založen na jazycích jako je Java a C++. Vzhledem k tomu, že se jedná o nativní jazyk, je u většiny uživatelů a v obchodě Unity Asset Store nejrozšířenější. Oproti psaní skriptů v UnityScriptu není nutné dělat dodatečné modifikace, a proto se stává nejoblíbenějším skriptovacím jazykem (Varga 2016).

3.4 Asset

Asset představuje jakýkoli prvek nebo soubor, kterým se dá projekt v Unity rozšířit. Jedná se například o skript, 3D model, obrázek, ikonu, zvukový soubor, prvek uživatelského rozhraní anebo libovolný soubor projektu. Assety mohou být vytvořené přímo v Unity jako např. skripty, prvek uživatelského rozhraní nebo v jiném programu jako třeba 3D modely, textury a zvukové soubory (Unity 2019).

3.5 Panely

Editor má několik panelů, přičemž každý z nich slouží ke svému specifickému účelu a je volně pohyblivý, takže umožňují vlastní přizpůsobení rozložení editoru. Hlavní z těchto panelů jsou Scene, Hierarchy, Inspector, Game, Project, Animator a Console, které budou blíže popsány. (Balcárek 2015).

Project

Panel obsahuje veškeré soubory spojené s aktuálním projektem uložené ve stromové struktuře složek. Jednotlivé složky by měly být správně pojmenovány. Např. Budeme-li chtít ukládat skripty, měli bychom vytvořit složku s názvem Scripts.

Hierarchy

Stromová struktura objektů (tzv. GameObjectů), kterým se dají přiřadit různé komponenty. Jedná se o uspořádání objektů ve scéně.

Inspector

Tento panel zobrazuje podrobné informace o aktuálně vybraném objektu a jeho aktuálně přiřazených komponentách a umožňuje jejich další přiřazení.

Scene

V tomto okně se upravuje a nahlíží do otevřené scény, přidávají se objekty, nastavuje se pozice, natočení a velikost. Ve scéně umožněn neomezený pohyb pro snazší editaci vybraných objektů.

Game

Panel Game zobrazuje scénu z pohledu hlavní kamery. Při spuštění scény se dostaneme do tzv. play módu, kde můžeme scénu otestovat a vidět, jak bude vypadat výsledně při sestavení.

Animator

Animator slouží k nastavení animací pro objekty. Pomocí časové osy lze měnit stav objektu v průběhu času.

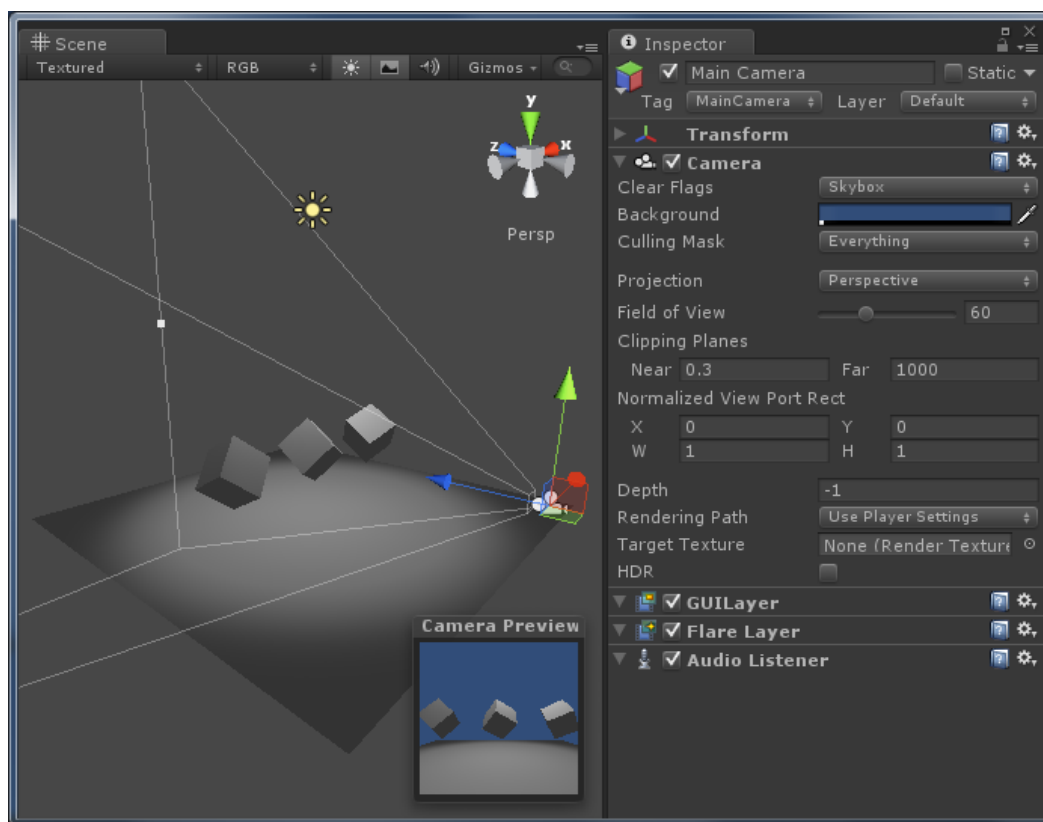
Console

Console vypisuje chyby a další informace do svého okna. Zobrazuje je jak při běžící aplikaci, tak při nalezení syntaktické chyby ve skriptech. Usnadňuje testování a hledání chyb.

3.6 Komponenty

Komponenta je funkční část každého GameObjectu a přiděluje mu jisté chování, které sama definuje. Právě díky komponentám a scriptům, které se mohou chovat také jako komponenty, získávají objekty určité vlastnosti a funkce. Unity má v základu spoustu komponent od transformací po řízení fyziky jednotlivých objektů

ve scéně. Zde jsou popsány hlavní základní komponenty, se kterými se uživatel setká nejčastěji.



Obrázek 3: Objekt s přiřazenými komponentami (Jeremiah van Oosten 2012)

Transform

Transform je základní komponenta, která je součástí každého GameObjectu a řídí veškeré transformace objektu. Umožňuje nastavit pozici vůči počátku scény, natočení a velikost.

Rigidbody

Slouží pro aplikaci fyzického chování objektů scény. Objekt s touto komponentou reaguje na gravitaci podle nastavení jeho váhy. Chování při kontaktu s jiným objektem je možné nastavit přiřazením fyzického materiálu.

Collider

V Unity existuje spousta druhů Colliderů jako Box Collider, Sphere Collider nebo Mesh Collider. Tyto komponenty GameObjectu nastaví jeho kolizní tvar a zabraňují protínání se s jiným objektem s touto komponentou.

Mesh Renderer

Mesh renderer umí graficky zobrazit objekt ve scéně. Pro zobrazení je nutné přiřadit komponentě takzvaný mesh, což je základní grafické primitivum, které tvoří největší část světa ve scéně.

Camera

Camera umí zobrazovat pohled do scény v herním okně a úzce spolupracuje s komponentou Transform, která kameře nastavuje, z jaké pozice a kterým směrem má scénu snímat.

Vlastní script

Každý script, který dědí ze třídy MonoBehaviour se dá také přiřadit jako komponenta. To umožňuje objektům přidávat vlastní chování.

3.7 Služby a vnitřní systémy

Unity3D disponuje velkým množstvím služeb. Mezi nimi se nachází statistiky, snadná implementace her pro více hráčů, cloudové sestavení projektu, nebo verzovací systém.

3.7.1 Unity Collaborate

Unity Collaborate je systém pro usnadnění práce v týmu. Jde o distribuční verzovací systém, který umožňuje uložení svého pracovního postupu do cloudových služeb a poskytnout je ostatním členům týmu spolupracujících na projektu. V případě, že tvůrce není členem žádného týmu, má alespoň přístup k verzím svého projektu a má možnost vidět, jak se projekt měnil v průběhu času.

3.7.2 Unity In-App Purchase

Je jednou z těchto služeb sloužících ke snazšímu zpeněžení. Jejím smyslem je snadné uvedení mikrotransakcí do funkčního stavu. Umožňuje uživatelům nakupovat virtuální měnu nebo herní prvky za měnu reálnou.

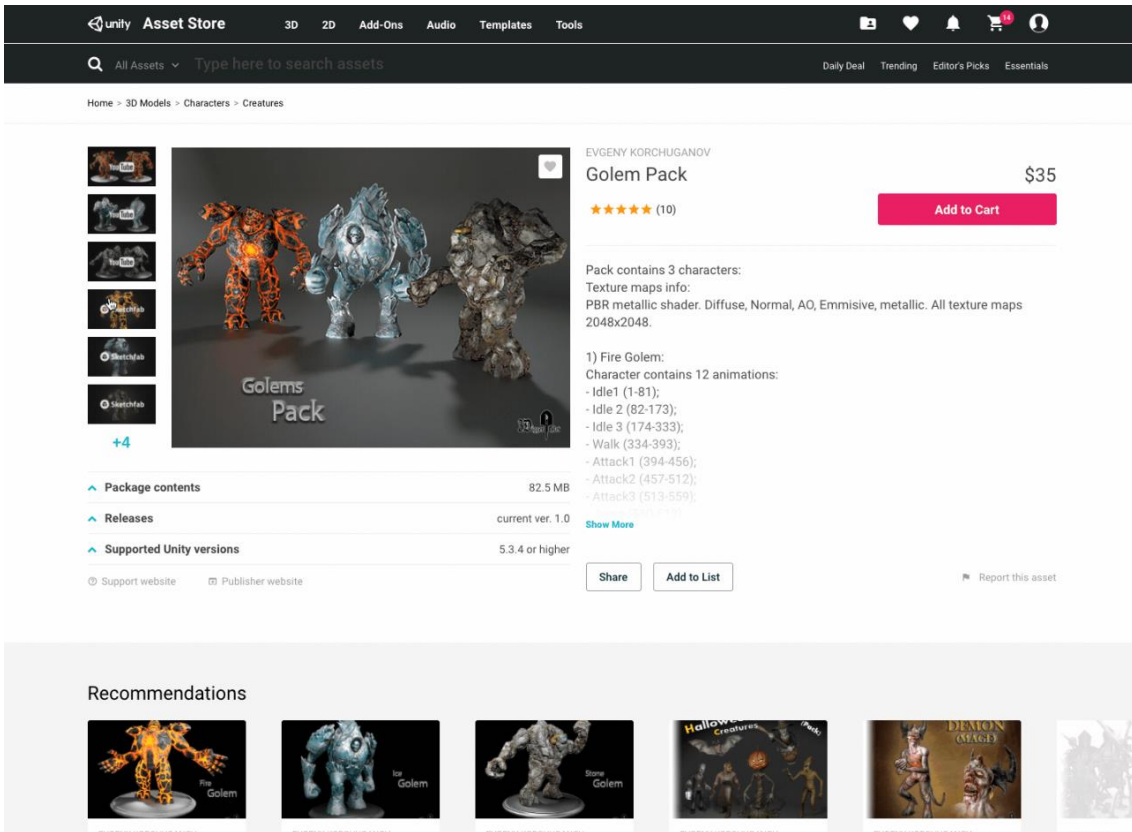
3.7.3 Unity Ads

Další podobnou službou je Unity Ads. Jejím primárním cílem je generovat tvůrci aplikace zisk, skrze zobrazování relevantních reklam uživatelům této aplikace. Zisk je určen podle množství zobrazených reklam jednotlivými uživateli a několika dalšími faktory.

3.7.4 Unity Asset Store

Unity Asset Store je obchod, kde jsou k dostání různé assety ke stažení. V obchodě se nachází mnoho placených assetů, z nichž jsou některé k dostání zdarma. Obvykle se cena odvíjí od kvality nabízeného assetu. K nalezení jsou zde assety, jak od členů komunity, tak od společnosti Unity Technologies. Do obchodu může nahrávat assety každý, kdo má účet pro Unity3D a asset, který chtěl publikovat, mu správci Asset Store schválí (Mesarč 2017).

Jedná se o skvělé místo k sehnání různých funkcionalit a rozšíření pro projekt. Získávání assetů tímto způsobem může velmi ušetřit čas.



Obrázek 4: Unity Asset Store (Nathan Ventura 2017)

4 Definice cílů a metodika zpracování

Na základě rešerše popsané v předchozích kapitolách jsou konkretizovány cíle práce a na jejich základě bude navržen a implementován praktický projekt realizující popsané přístupy.

4.1 Cíl systému

Cílem je odzkoušení postupů tvorby aplikace za pomoci softwarového nástroje Unity3D využívající systému virtuální reality HTC Vive. Bude se jednat o interakci uživatele uvnitř indoorové prostorové scény s možností vkládání informací z dalších datových zdrojů. Uživatel bude mít k dispozici dva ovladače, které budou viditelné ve vizualizaci a prostřednictvím snímáním stavu jejich tlačítek bude umožněna interakce s objekty ve scéně. Těmito ovladači bude možné zobrazit laserové ukazovátko sloužící k pohybu po scéně nebo ke komunikaci s uživatelským rozhraním, které bude možné si tlačítkem zobrazit. Toto ukazovátko bude získávat informace o objektu, na který aktuálně míří a závisle na získaných informacích měnit způsob jeho zobrazení. V případě, že laser bude ukazovat na panel s umožněnými interakcemi nebo objekt, po kterém je umožněn pohyb, jeho barva se změní.

Pro testování bude využit 3D model představující budovu Fakulty informatiky a managementu včetně vnitřního uspořádání učeben a dalšího vybavení interiéru. Doplňující informace znázorněné ve scéně v textové i vizuální podobě budou získávány z informačního systému fakulty STAG. Pro přístup k datům bude využita nová REST služba postavená nad tímto systémem sloužící právě k získávání dat. Výsledně navrácená data poskytující informace o aktuálních rozvrzích jednotlivých učeben budou zobrazena vedle dveří patřičných učeben. K učebnám bude možnost přesunu formou teleportu nebo pomocí automatické navigace, které se předá místnost pomocí panelu zobrazeného ovladači.

4.2 Výběr technologií

Jednou z technologií užitých při vývoji aplikace bude herní engine Unity3D, který je zvolen z důvodů podpory široké škály zařízení, pro které výslednou aplikaci může sestavit, intuitivnímu prostředí, skvělé podpoře virtuální reality a programovacího jazyka C#. Další výhodou je možnost využití modelu interiéru

budovy Fakulty informatiky a managementu (FIM) vytvořeného v rámci diplomové práce Marka Pěničky (Pěnička 2012), která byla rovněž tvořena v Unity3D.

Využitým zařízením zobrazující virtuální realitu pro tvorbu aplikace bude HTC Vive. Jde o populární a rozšířenou technologii, která je velmi dobře podporována vybraným herním enginem. Má k dispozici dva dobíjecím ovladače, které mohou umožnit interakce uvnitř virtuální scény. A pro získávání informací o směru natočení headsetu a ovladačů využívá dvou statických základních stanic napájených z elektrického zdroje.

Jako zdroj dat do scény bude využito nového univerzitního REST API, které je postavené nad informačním systémem STAG. Tento zdroj dat byl zvolen za účelem získání dat, která jsou logicky spojená s budovou FIM. A také pro jeho jednoduchou komunikaci přes HTTP protokol.

Tabulka 1: Specifikace HTC Vive (VIVE™ 2019)

Obraz displaye	Dual AMOLED 3.6" diagonal
Rozlišení	1080 x 1200 pro každé oko
Snímková frekvence	90 Hz
Šíře zorného pole	110 stupňů
Senzory	SteamVR Tracking, G-sensor, gyroskop, senzor blízkosti
Připojení	HDMI, USB 2.0, 3.5 mm jack, Napájení, Bluetooth



Obrázek 5: Headset HTC Vive (HTC Corporation 2019)

5 Zdroje dat do scény

Za zdroj dat se může považovat kterýkoli další soubor nebo systém poskytující data nebo soubory jako jsou modely objektů, textury a textové soubory různých formátů. Tyto systémy se nazývají aplikační programové rozhraní (API) a umožňují přes jejich metody z veřejného rozhraní přistupovat k datům, která poskytují.

5.1 API

API je anglická zkratka pro Application Programming Interface používaná v počítačovém programování a vývoji webových aplikací. Přičemž interface neboli rozhraní je způsob, jakým lidé komunikují s programy. Většina programů, s nimiž lidé komunikují, mají grafické nebo vizuální uživatelské rozhraní, které se skládá z oken s tlačítky pro odesílání konkrétních příkazů (Bampende 2015).

Rozhraní API lze chápat jako bod interakce pro aplikaci, která chce použít funkce jiné aplikace za účelem provedení určitého úkolu. API bývají vystavovány vývojáři za účelem poskytování služeb dalším vývojářům, kteří dále mohou využívat tyto rozhraní ve svých systémech. Tyto služby poté mohou být zdarma anebo je nutno si je zakoupit.

Jednoduše řečeno se jedná o programové rozhraní, které umožňuje programu komunikaci s jinými systémy. Je to soubor předdefinovaných metod napsaných v některém z programovacích jazyků, který má obvykle neznámou vnitřní logiku k provedení určitých úkolů. Na rozdíl od lidí, aplikace nepotřebují uživatelské grafické prostředí pro přístup k používaným metodám nebo funkcím (Bampende 2015).

5.2 JSON

JSON (JavaScript Object Notation) je formát pro výměnu dat, který je nezávislý na počítačové platformě. Nejčastěji se používá k přenosu dat, která mohou být obsažena v polích nebo objektech. Jedná se o jednodušší alternativu k XML. JSON má čitelnější strukturu, je úspornější a lépe se zpracovává.

XML

```
<empinfo>
  <employees>
    <employee>
      <name>James Kirk</name>
      <age>40</age>
    </employee>
    <employee>
      <name>Jean-Luc Picard</name>
      <age>45</age>
    </employee>
    <employee>
      <name>Wesley Crusher</name>
      <age>27</age>
    </employee>
  </employees>
</empinfo>
```

JSON

```
{ "empinfo" :
  {
    "employees" : [
      {
        "name" : "James Kirk",
        "age" : 40,
      },
      {
        "name" : "Jean-Luc Picard",
        "age" : 45,
      },
      {
        "name" : "Wesley Crusher",
        "age" : 27,
      }
    ]
  }
}
```

Obrázek 6: Porovnání JSON a XML (Hoi 2017)

5.3 IS/STAG

IS/STAG je zkratka pro informační systém studijní agendy. Jedná se o webovou aplikaci obsahující řadu webových služeb poskytujících data týkajících se univerzity, kterou je systém využíván. V informačním systému je k dispozici množství aktualizovaných dat o vyučovaných předmětech a oborech, učitelích, studentech, jejich rozvrzích ale i jednotlivé známky předmětů určitých studentů. Tento informační systém ve většině případů pokrývá funkce od přijímacího řízení až po vydání diplomu.

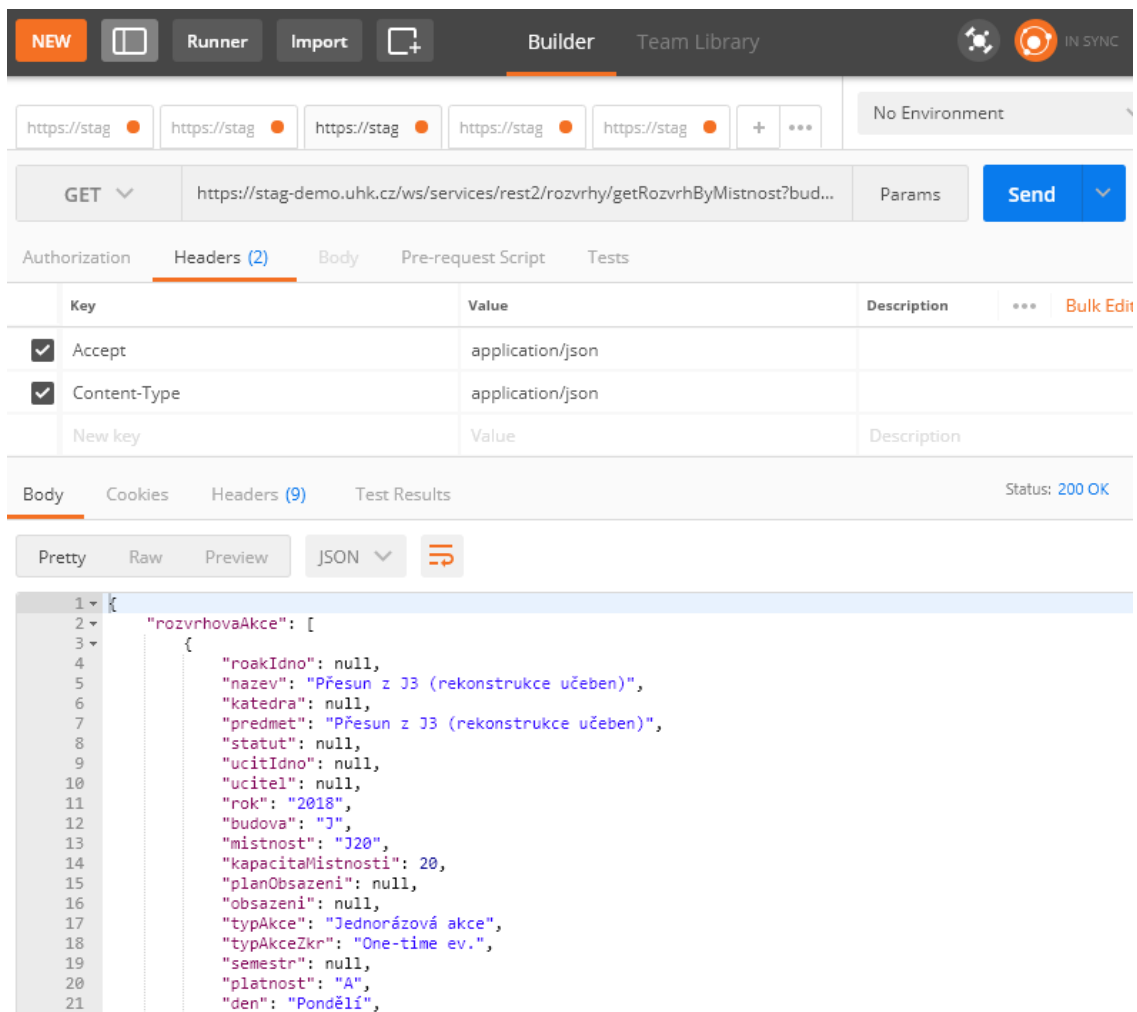
5.3.1 REST rozhraní

Representational State Transfer je styl pro návrh síťových aplikací založený na modelu komunikace klient-server, u kterého jsou veškerá data uložena na straně serveru, a přistupuje se k nim z klientského zařízení pomocí HTTP protokolu. Přístup k datům se uskutečňuje pomocí URL, která z klientského zařízení posílá požadavek přes HTTP protokol směřující na konkrétní server. Server tento

požadavek přijímá, zpracuje ho a navrací zpět informace o úspěchu či neúspěchu a v případě úspěchu i výstupní data požadovaná původním voláním.

REST využívá standardní CRUD operace pro manipulaci s daty uloženými na serveru. Kde pomocí HTTP metod, GET, POST, UPDATE a DELETE můžeme manipulovat s daty. Prostřednictvím těchto metod můžeme data ze serveru číst, posílat nebo přidávat na něj nová, anebo také je ze serveru mazat. HTTP protokol umožňuje také definování klientských hlaviček, které nám například umožňují zaslat informace o zasílaném nebo požadovaném datovém formátu.

Výstupní data se dají jistým způsobem ovlivnit pomocí parametrů zasílaných společně s požadavkem, pokud to volaná operace umožňuje. Tyto parametry pak mohou být dvojího typu. Jedněmi z nich jsou vyžadované parametry, které je nutné zadat, aby volání operace neskončilo selháním. A druhými jsou volitelné parametry, u kterých si můžeme vybrat, zda je chceme zadávat. Pak se při jejich vyplnění se musí shodovat požadovaný formát se zadaným. Tzn., Pokud operace umožňuje nebo dokonce vyžaduje parametr „početKusů“ číselného typu, na místě hodnoty by se měli nacházet pouze numerické znaky (0-9).



Obrázek 7: Příklad volání metody s parametry v prostředí Postman

6 Návrh a implementace systému

Obsah této kapitoly se zabývá samotným postupem návrhu a tvorby finálního systému. Uvedeny jsou hlavní kroky postupu práce na projektu týkajících se vytvoření projektu, kdy bylo třeba získat a upravit model budovy FIM, umožnit pohyb a nastavit automatické navigace, řešit naskytnuté problémy a finálně načíst z datového zdroje data s rozvrhy a zobrazit je ve scéně.

6.1 Vytvoření projektu

Jeden z prvních úkolů se věnoval založení nového projektu v Unity3D. Šablonou projektu byla zvolena LWRP (angl. Lightweight Render Pipeline) pro VR. Jde o zjednodušený postup vykreslování herního prostředí za pomoci materiálů využívajících nenáročné shadery.

Následuje export souborů s assety z projektu obsahující již vytvořený objekt budovy Fakulty informatiky a managementu (Pěnička 2012) a veškeré související textury a materiály. Tyto soubory se naimportovaly do současného projektu a za pomoci zabudovaného verzovacího systému Collaborate, zmíněného v kapitole o herním enginu Unity3D, vytvořit první cloudově zálohovanou verzi celého projektu.

6.2 Úprava scény

Příprava scény se skládala ze dvou kroků, upravení transformace budovy FIM a převedení materiálů z předešlé práce (Pěnička 2012) z důvodu kompatibility s verzí současnou. Jelikož předchozí projekt byl tvořen v mnohem starší verzi Unity3D, ve které byly jinak orientovány souřadnicové osy, následovalo správné natočení objektu a uvedení do správného měřítka v aktuálním souřadnicovém systému. Z důvodů zastaralosti materiálů, což jsou grafické vlastnosti objektů např. textury, použitých v předchozím projektu, které již nejsou nadále podporovány, bylo nutné materiály konvertovat a v některých případech vytvořit nové.



Obrázek 8: Model budovy FIM v Unity3D

6.3 Hierarchické dělení tříd

Pro zvýšení přehlednosti a kontroly nad provázaností výsledných tříd je udržováno jejich dělení podle smyslu jejich využití. Třídy jsou děleny hierarchicky do složek, tak aby v každé složce byly jen ty, které budou mít za úkol stejnou nebo podobnou činnost.

Model

V této složce se nachází třídy pro získávání dat z univerzitního systému STAG. Třídy slouží pouze jako držitelé logicky souvislých dat. V některých případech nemusí načítat veškeré získané údaje a ukládají si pouze ty, které potřebují pro svůj účel využití.

UI

User interface (UI) jsou třídy pracující s prvky uživatelského rozhraní. Mají za úkol spravovat k sobě připojené komponenty, se kterými pracují. Ve většině případů se jedná o vstupní textová pole, posuvníky, texty a obrázky, kterým pouze nastavují obsah nebo manipulují se vzhledem.

Controllers

Cílem controllerů je volat metody provázaných komponent nebo tříd sloužících pouze pro správu velkého počtu komponent najednou. Obvykle jsou propojeny s UI třídami, kterým delegují činnosti, které mají provést.

Managers

Veškeré třídy s návrhovým vzorem singleton jsou uloženy v této složce. Mají největší provázanost na ostatní třídy a zavoláním jejich metod se pouze delegují metody na provázaných třídách, kterými ve většině případů bývají controllery. Jejich primárním smyslem je umožňovat volání jejich veřejných metod kdekoli jinde v kódu aplikace skrze přistoupení ke statické instanci.

6.4 Pohyb ve scéně

Jedním z řešených problémů praktické části práce, byla implementace logiky pro umožnění pohybu ve virtuálním prostředí. Problém je spojen se situací při fyzickém pohybu uživatele VR, kdy je jeho reálný pohyb snímán základními stanicemi a promítán do scény. Zařízení virtuální reality umožňuje pohyb pouze v omezeném rozsahu. Ten je limitován maximálním dosahem snímacích stanic a fyzickým prostorem místnosti. Pro tuto situaci je k dispozici následujících několik řešení. Do konečného projektu byly vybrány dvě implementace oblíbené z důvodu jejich praktického použití.

6.4.1 Teleport skokem v daném směru

Myšlenka skokového pohybu virtuálním prostorem spočívá v přesunu ve směru, kam ukazuje ovladač, na kterém došlo ke stisknutí tlačítka zodpovědného za spuštění metody pohybu. Metoda potřebuje jako vstupní parametry pouze směr požadovaného pohybu, který je získán ze směru natočení ovladače a pevně nastavenou vzdálenost o kterou se posuneme. Při volání metody dojde k vypočtení souřadnic výsledné pozice a přesunu kamery představující uživatele na vypočtenou pozici. Tento pohyb byl také doprovázen rychlým zatměním obrazu, pro eliminaci negativních účinků na osobu s nasazeným headsetem.

6.4.2 Teleport na danou pozici

Způsob pohybu formou teleportu využívá fyziku z Unity3D, konkrétně metodu Raycast, která potřebuje jako parametry pozici, směr, vzdálenost a informace o tom jaké objekty má při zavolání zohledňovat. Funguje principem vedení paprsku s využitím zadaných parametrů. Metoda poté navrací data s informacemi, zda zasáhla nějaký objekt. V kladném případě vrátí informace o zásahu a současně i pozici zásahu kam chceme přesunout náš virtuální charakter. Po získání výsledné pozice dojde k přesunutí.

6.4.3 Posun daným směrem

Jedná se o myšlenku velmi podobnou teleportu skokem v daném směru, s rozdílem, že do výsledné pozice se charakter přesouvá v průběhu krátkého úseku času a nedochází k bleskovému přeskoku na vypočtenou pozici. I přes to, že tento způsob hodně blíží reálnému pohybu, uživatelsky byl nejméně přívětivý. Nejčastěji způsoboval nepohodlí, dezorientaci a ztrátu stability uživatele spojenou s pojmem cybersickness. Tyto symptomy způsoboval už známý problém, kdy uživatel stojí pevně na svém místě, kdežto ve virtuálním světě je v pohybu.

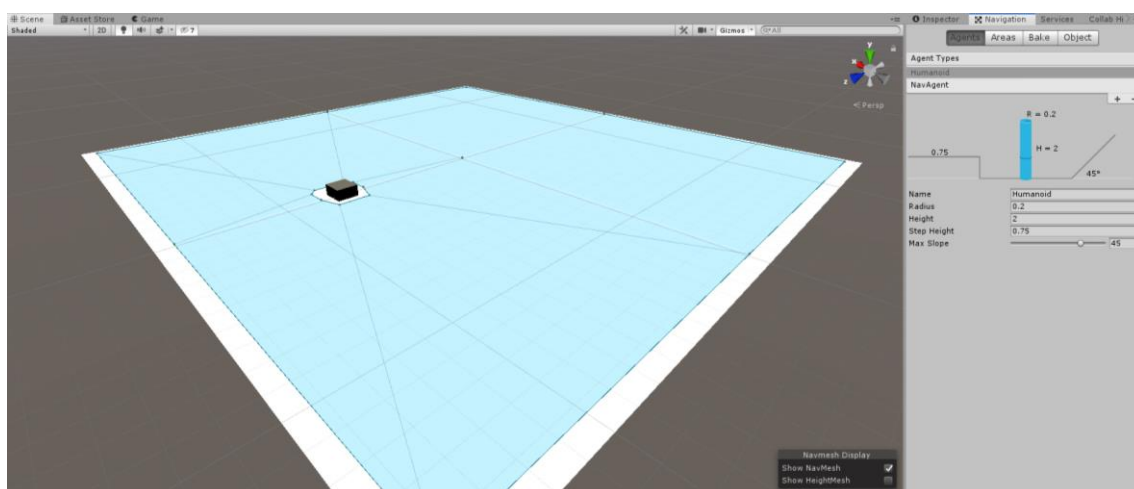
6.4.4 Automatická navigace

Pro zprovoznění navigace uvnitř budovy je využito navigačního systému s umělou inteligencí. Tento systém se neustále vyvíjí a je jako balíček zakomponován v Unity3D.

Navigační systém obsahuje několik komponent k usnadnění automatizovaného pohybu ve virtuálním prostředí. V aplikaci jsou využity dvě komponenty nutné k samotnému fungování navigací. Jedna z nich je NavMesh, který je nutno vygenerovat pro správnou funkčnost navigací. Při generování se definují objekty, po kterých má a nemá být umožněn automatický pohyb. Výstupem generování je objekt určující plochy s umožněným automaticky řízeným pohybem.

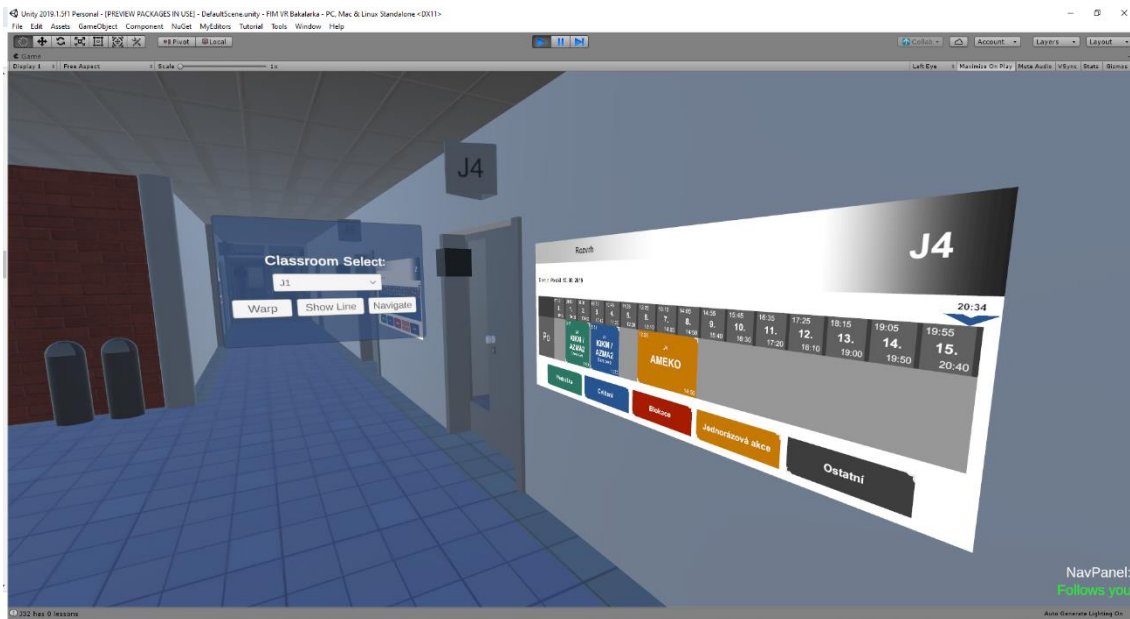
Druhou komponentou je NavMeshAgent, která se přidá našemu charakteru, nastaví se jí vhodně potřebné atributy jako je šířka, výška a rychlost pohybu. Komponenta řídí veškerý pohyb objektu, se kterým je spojena. K dosažení pohybu stačí zavolat metodu SetDestination komponenty NavMeshAgent, které je předán

vstupní parametr s požadovanou pozicí a komponenta začne charakter autonomně přemísťovat k cíli.



Obrázek 9: Příklad vygenerovaného NavMesh objektu

Aplikace využívá výše uvedené komponenty pro umožnění pohybu ve scéně. Ve virtuálním prostoru se před každou učebnou nachází skryté objekty. Každý z těchto objektů je propojen s názvem učebny. Zobrazením uživatelského rozhraní pomocí ovladačů VR a vybráním jedné z možných učeben se zavolá metoda spravující automatickou navigaci. Ta nalezne pomocí identifikátoru odpovídající objekt odkazující na místo učebny a předá se 3D vektor s jeho pozicí metodě komponenty NavMeshAgent, která vykoná výslednou navigaci na pozici.



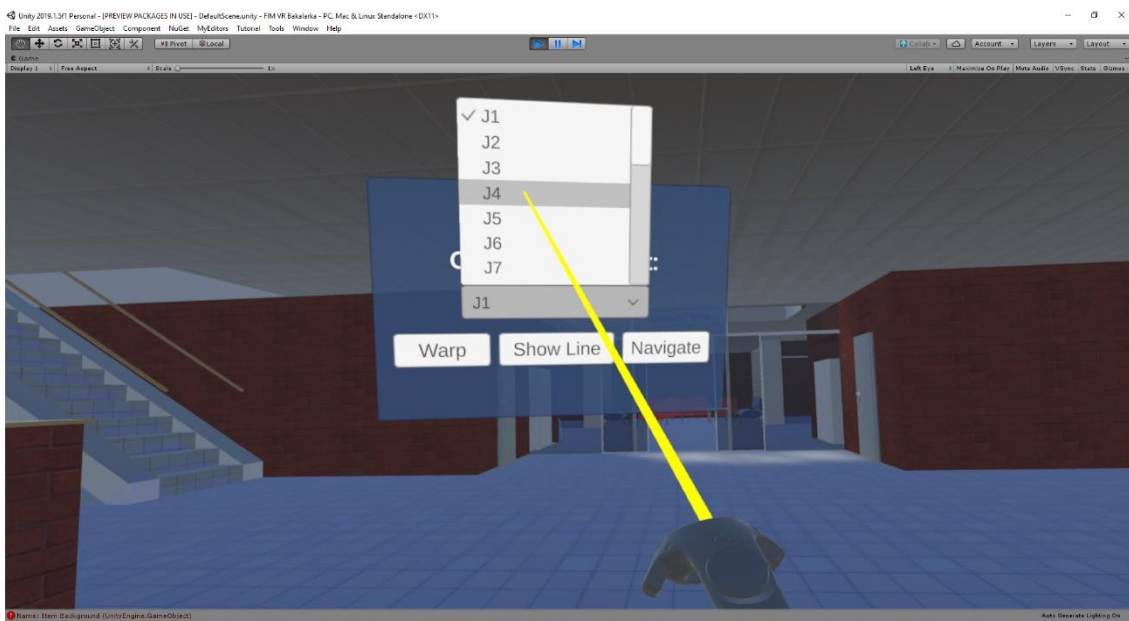
Obrázek 10: Učebna se svým rozvrhem a UI pro navigace

6.5 Interakce

Interakce jsou umožněny prostřednictvím ovladačů VR. Zpřístupňují uživateli možnost pohybovat se po virtuální scéně pomocí teleportu nebo si zobrazit uživatelské rozhraní pro výběr místnosti, ke které se chce uživatel nechat dovést automatickou navigací.

Princip fungování pro zjištění stisku tlačítka ovladače je velmi jednoduchý. V každém snímku se kontroluje, zda nedošlo ke stisknutí požadovaného tlačítka na vybraném ovladači. V kladném případě se zavolá metoda vyčkávací na stisk tohoto tlačítka.

Větší komplikace nastávají u interakcí s uživatelským rozhraním, které k interakci vyžaduje kameru a pozici kurzoru myši. 2D Souřadnice pozice myši jsou aplikovány na obraz kamery, ze které je vyslán neviditelný paprsek kontrolující, zda nezasáhl nějaký z objektů umožňující interakce. Tento princip využívá pouze jednu kameru pro uživatelské rozhraní. V důsledku tohoto faktu a při použití této implementace lze umožnit pouze jednomu ovladači interakce s uživatelským rozhraním.

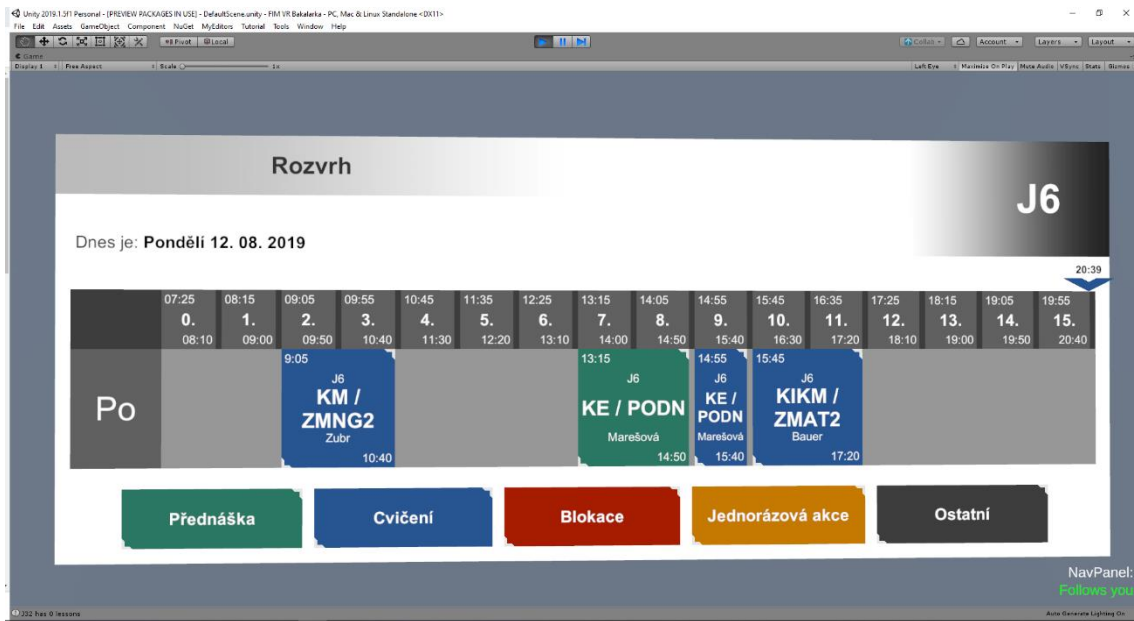


Obrázek 11: Příklad interakcí ve VR prostřednictvím laserových ukazovátek

6.6 Integrace externích dat do scény

Data do scény jsou získávána asynchronní metodou, která pro zavolání požaduje URL, sestavenou z parametrů odkazující se na REST API univerzity. Jako druhý parametr je předávána callback metoda, které se po úspěšném získání dat tyto data předávají jako parametr.

Uvnitř asynchronní metody dochází pouze k nastavení hlaviček pro požadavek a zaslání požadavku na server. Předaná callback metoda vyčkává na její zavolání a dále si zpracovává data po svém. V tomto případě předaná data namapuje na instanci modelové třídy Rozvrh a předává je další metodě. Ta jednotlivé hodiny rozvrhu seřadí podle času, kdy začínají a z každé poskládá grafický objekt, který je následně přidán do rozvrhu a ten je zobrazen vedle dveří patřičné místnosti.



Obrázek 12: Příklad vygenerovaného rozvrhu ze získaných dat

7 Výsledky

Implementace aplikace nečinila žádné velké problémy, až na pár komplikací zmíněných dále. Získání modelu budovy a potřebných souborů proběhlo v několika jednoduchých krocích. Bylo pouze nutné konvertovat zastaralé soubory z předešlého projektu pro zaručení funkčnosti.

Zajištění interakcí ve VR činilo větší potíže. Tyto potíže byly spojeny s logikou řízení událostí uživatelského rozhraní, kterou využívá Unity3D. Pro uživatelské rozhraní je možnost vybrat jednu kameru, pro kterou se zjišťuje aktuální pozice kurzoru v okně aplikace, a kontroluje se, zda kurzor není nad objektem, který umožňuje interakce. Tohoto principu bylo využito i v tomto případě, kdy pouze na jeden z ovladačů byla přidána komponenta kamery s pevně nastaveným kurzorem na její střed a je předána uživatelskému rozhraní, pro dosažení požadovaného výsledku.

Při implementaci došlo k několika kritickým bodům. Často se jednalo o chyby vzniklé nevhodným návrhem provázanosti tříd, programové chyby nebo neočekávané situace. Příkladem nevhodného návrhu je nepřístupnost atributu či metody ve třídě, která s nimi potřebovala pracovat. Tento případ měl snadné řešení. Naopak v případě neočekávaných situací došlo při implementaci automatické navigace k situaci, kdy byl uživatel přesouván skrze zdi.

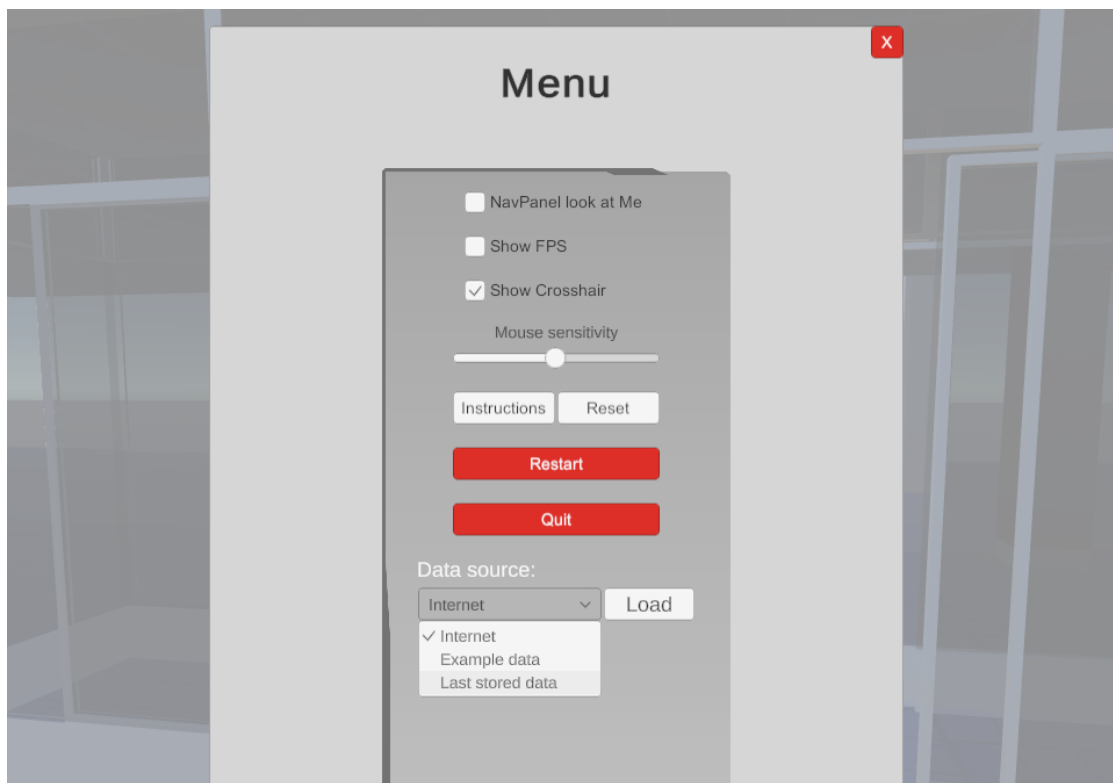
Problém byl způsoben pohybem v hierarchii nadřazeného objektu (tzv. Rigu), který ve virtuální scéně zobrazuje hranice reálného prostoru, pro který je nastaven pohyb. Rig slouží pro varování ohledně vystoupení z herní zóny a z dosahu základních stanic - v reálném prostředí může dojít ke kolizi s nějakým objektem. Potomkem Rigu je uživatel VR (kamera ve scéně), ten se pohybuje relativně vůči tomuto objektu s hranicemi. Po přidání komponenty NavMeshAgent (pro automatické navigace) na Rig tato komponenta pohybovala a rotovala středem Rigu. V tomto případě mohla nastat situace, kdy je uživatel posunut vůči středu Rigu a může být veden zdmi.



Obrázek 13: Příklad objektu Rig ve scéně SteamVR Home

Řešením problému s procházením zdmi při automatické navigaci bylo synchronizování transformací kamery a přidáním dalšího rodiče nad objekt Rigu. Na tento objekt se umístila navigační komponenta a v každém snímku se tomuto objektu nastavovala pozice a rotace kamery. Posledním krokem bylo vrácení Rigu do stavu před nastavením transformací rodičovského objektu, jehož tyto transformace ovlivnily. Bylo dosaženo stavu, kdy uživatel spustí automatické navigování a je navigován směrem, kterým se dívá. K řešení byla navíc přidána funkce zobrazení čáry směřující z aktuální pozice na pozici zadanou v uživatelském rozhraní.

do lokálního adresáře po detekci internetového připojení. Z adresáře se načítají data v případě nedostupnosti internetového připojení. V adresáři je k nalezení složka, která obsahuje testovací data a ta jsou do scény vkládána při nedostupnosti připojení a zároveň při problému načtení lokálních dat.



Obrázek 16: GUI pro desktopovou verzi s možností volby zobrazovaných dat

Aby bylo možné poskytnout aplikaci k otestování bez fyzického zařízení pro virtuální realitu, bylo potřeba vytvořit nové uživatelské rozhraní, které nahradilo původní rozhraní určené pro VR. To bylo nutné z důvodu používání rozdílných přístupů řešení interakcí. S tím souviselo přepsání metod, které s cílem provedení stejné úkolu vykonávaly rozdílné činnosti.

Výsledně se veškeré tyto komplikace podařilo vyřešit a poskytnout vyzkoušení kolegům a lidem účastnících se prezentačních akcí na FIM UHK. Na základě jejich hodnocení byl přijat závěr potvrzující oblíbenost této technologie u převážné většiny uživatelů. V některých individuálních případech nastával stav, kdy si uživatel nasadil zařízení na hlavu a stěžoval si na nepohodlí a jistou míru dezorientace.

8 Závěry a doporučení

Záměrem této bakalářské práce bylo prozkoumání možností implementace virtuální reality v programu Unity3D a na základě získaných informací zhotovit praktickou část. Praktická část se má skládat z aplikace umožňující získání dat z externích zdrojů a poskytnutí interakcí ve virtuální scéně.

Myšlenka technologie virtuální reality spočívá v utvoření co nejvěrohodnější simulace reálného světa jejímu uživateli prostřednictvím jejích brýlí, které promítají stereoskopický obraz. Jejím hlavním úkolem je ovlivnění smyslů jako je zrak sluch a v některých případech i hmat. V současnosti je technologie VR mnohem více rozšířena mezi běžné uživatele. Nabízí uplatnění ve spoustě průmyslových oblastí. Ale může mít negativní dopad na zdraví jejího uživatele spojený s pojmem „virtual reality sickness“.

Pro praktickou realizaci byl použit herní engine Unity3D, což je komplexní integrované vývojové prostředí. Jeho jádro poskytuje základní funkce jako je vykreslování, výpočty fyziky, detekce kolizí a mnoho dalších. Příkladem je hra DOOM, která měla oddělenou vlastní náplň hry od samotného jádra. Unity3D nabízí širokou škálu podporovaných platforem, pro které se dá aplikace sestavit. Tvoření aplikací je zprostředkováno objekty tzv. „GameObject“, které mají vlastní transformaci a umísťují se na ně různé komponenty, které objektu přidávají chování.

Prostředím umožňujícím interakci v modelované scéně byl model budovy FIM s přidanými informacemi získaných z externích zdrojů. Zdrojem dat do scény byl školní informační systémem STAG, přístupný prostřednictvím REST API. Rozhraní je založeno na modelu klient-server a využívá HTTP protokol ke komunikaci. Uživateli je umožněn volný pohyb po interiéru budovy FIM, s možností dvou typů přesunu, teleport a automatická navigace. Před jednotlivými učebnami jsou zobrazovány aktualizované rozvrhy získané z informačního systému STAG.

Během zhotovování požadované aplikace se vyskytly následující komplikace: Úprava projektu pro testování aplikace bez fyzického zařízení pro virtuální realitu, vyřešení správné funkcionality automatické navigace, mapování dat ve formátu

JSON na modelové třídy, špatná logika pro zobrazování dat na rozvrhu a implementace chování aplikace při nedostupnosti internetového připojení.

Problémy se v průběhu tvorby povedlo vyřešit a výsledně sestavit aplikaci. V průběhu realizace byly jednotlivé vývojové verze testovány a využívány v rámci prezentačních akcí FIM UHK, jako je Den otevřených dveří a Noc vědců. Ukazuje se, že systém Unity 3D v kombinaci se zařízením HTC Vive tvoří vhodné prostředí pro vývoj aplikací pro virtuální realitu.

Navázáním na tuto práci by se v rámci dalších prací mohla aplikace rozšířit o funkci automatického navigování za osobou vyučujícího. Tato funkce by spočívala ve zjištění informace, zda má vyučující v daném čase konzultace a zda se nachází ve své kanceláři. V případě splnění těchto podmínek by automatická navigace nasměrovala uživatele k jeho kanceláři. V opačném případě by se zjistilo, zda se vyučující nenachází v některé z učeben, kam by byl uživatel navigován. Nabízela by se i další možnost prozkoumání integrace více hráčů za využití virtuální reality.

Seznam použité literatury

- © MICROSYS, SPOL. S R. O., 2016. JSON - textový formát pro výměnu dat. *PROMOTIC* [online] [vid. 2019-04-07]. Dostupné z: <https://www.promotic.eu/cz/pmdoc/Appendix/FileFmt/json/FmtJSON.htm>
- BALCÁREK, Daniel, 2015. *Virtuální svět* [online] [vid. 2019-08-13]. Brno. Diplomová práce. VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ. Dostupné z: https://dspace.vutbr.cz/bitstream/handle/11012/41710/DP_Daniel_Balcarek.pdf?sequence=-1
- BAMPENDE, Humberto Hetu, 2015. *Third Party Web API Implementation in Web Application Development*. Bakalářská práce. Helsinki Metropolia University of Applied Sciences.
- DANSIE, Jason, 2013. *Game Development in Unity* [online] [vid. 2019-08-13]. Helsinki. Bakalářská práce. Helsinki Metropolia University of Applied Sciences. Dostupné z: https://www.theseus.fi/bitstream/handle/10024/68068/Dansie_Jason.pdf
- HTC CORPORATION, 2019. *HTC Vive* [online] [vid. 2019-08-13]. Dostupné z: https://www.vive.com/media/filer_public/vive/product-overview/vive-hardware-hmd-1.png
- JUN, Lukáš, 2017. *VYUŽITÍ VIRTUÁLNÍ REALITY V PROJEKCI A KONSTRUKCI VÝROBNÍCH STROJŮ, SYSTÉMŮ A ROBOTIKY* [online]. Brno. VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=149602
- KOVÁŘÍK, Stanislav, 2017. *Vývoj 3D her ve frameworku Unity* [online]. B.m. [vid. 2018-08-25]. Bakalářská práce. Vysoká škola ekonomická v Praze. Dostupné z: https://theses.cz/id/2qv1j5/61178_kovs00.pdf
- LOWOOD, Henry E., 2018. Virtual reality - Entertainment. *Encyclopedia Britannica* [online] [vid. 2019-08-13]. Dostupné z: <https://www.britannica.com/technology/virtual-reality>
- MESARČ, Jan, 2017. *Vývoj multiplatformní hry v Unity3D* [online]. Pardubice. Bakalářská práce. Univerzita Pardubice. Dostupné z: http://dspace.upce.cz/bitstream/handle/10195/68095/Mesarcj_VyvojMultiplatformni_RD_2017.pdf?sequence=1&isAllowed=y
- PĚNIČKA, Marek, 2010. *Modelování reálné prostorové scény*. Hradec Králové. Bakalářská práce. Univerzita Hradec Králové.
- PĚNIČKA, Marek, 2012. *Navigace v prostorové scéně*. Hradec Králové. Diplomová práce. Univerzita Hradec Králové.

SKALA, Lukáš, 2017. *Virtuální realita a její použití v herním průmyslu*. B.m. Bakalářská práce. Masarykova univerzita.

UNIFY COMMUNITY WIKI, 2014. *UnityScript versus JavaScript - Unify Community Wiki* [online] [vid. 2018-08-26]. Dostupné z: http://wiki.unity3d.com/index.php/UnityScript_versus_JavaScript

UNITY, 2019. Unity. *Unity* [online] [vid. 2019-08-13]. Dostupné z: <https://unity.com/frontpage>

UNITY TECHNOLOGIES, 2019. Unity Technologies. *Unity Store* [online] [vid. 2019-08-13]. Dostupné z: <https://store.unity.com/>

VARGA, Ervin, 2016. *Platforma Unity 5 – její vlastnosti a použití při vývoji 2D počítačové hry* [online]. Praha [vid. 2018-08-25]. Bakalářská práce. Vysoká škola ekonomická v Praze. Dostupné z: https://vskp.vse.cz/50636_platforma_unity_5___jeji_vlastnosti_apouziti_pri_vyvoji_2d_pocitacove_hry

VIRTUAL REALITY SOCIETY, 2017. History Of Virtual Reality. *Virtual Reality Society* [online] [vid. 2019-08-10]. Dostupné z: <https://www.vrs.org.uk/virtual-reality/history.html>

VIVE™, 2019. VIVE™. *HTC VIVE™* [online] [vid. 2019-08-13]. Dostupné z: <https://www.vive.com/eu/product/>

ZOE CLAYTON, 2013. Stereographs • V&A Blog. *V&A Blog* [online] [vid. 2019-08-10]. Dostupné z: <https://www.vam.ac.uk/blog/caring-for-our-collections/stereographs>

Oskenované zadání práce

Univerzita Hradec Králové
Fakulta informatiky a managementu
Akademický rok: 2018/2019

Studijní program: Aplikovaná informatika
Forma: Prezenční
Obor/komb.: Aplikovaná informatika (ai3-p)

Podklad pro zadání BAKALÁŘSKÉ práce studenta

PŘEDKLÁDÁ:	ADRESA	OSOBNÍ ČÍSLO
Bareš Tomáš	Hronova 1559, Náchod	I1600511

TÉMA ČESKY:

Virtuální realita v Unity3D

TÉMA ANGLICKY:

Virtual reality in Unity3D

VEDOUCÍ PRÁCE:

Ing. Bruno Ježek, Ph.D. - KIKM

ZÁSADY PRO VYPRACOVÁNÍ:

Cíl

Prostudovat problematiku systémů virtuální reality a jejich aplikace v počítačové grafice se zaměřením na import dat z vnějších zdrojů. S využitím technologie Unity3D vyzkoušet vybrané principy integrace dat do prostorové scény.

Osnova práce

1. Provést literární rešerši a seznámit se s aktuálním poznáním v oblasti virtuální reality.
2. Vytvořit přehled principů, technologií a aplikací využívaných v systémech virtuální reality.
3. Prozkoumat možnosti systému Unity3D pro modelování, interakci a zobrazení prostorové scény.
4. Zvolit vhodný příklad využití integrace dat do prostorové scény a navrhnout její softwarovou implementaci.
5. Pro implementované řešení provést testování a zhodnotit dosažené výsledky.

SEZNAM DOPORUČENÉ LITERATURY:

Unity. Unity [online] [vid. 2018b-10-22]. Dostupné z: <https://unity3d.com>

VIVETM | VIVE Virtual Reality System [online] [vid. 2018c-10-22]. Dostupné z: <https://www.vive.com/us/product/vive-virtual-reality-system/>

DANSIE, Jason, 2013. Game Development in Unity: Game Production, Game Mechanics and the Effects of Gaming [online]. [vid. 2018-10-22]. Dostupné z: <http://www.theseus.fi/handle/10024/68068>

Podpis studenta: Bareš

Datum: 22.10.2018

Podpis vedoucího práce: [Podepsano]

Datum: 22.10.2018