



Zadání bakalářské práce

Název:	Metoda loci ve virtuální realitě
Student:	Dominik Pupala
Vedoucí:	doc. Ing. Mgr. Petr Klán, CSc.
Studijní program:	Informatika
Obor / specializace:	Webové a softwarové inženýrství, zaměření Počítačová grafika
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	do konce letního semestru 2021/2022

Pokyny pro vypracování

Implementujte paměťovou metodu loci ve virtuálním 3D prostředí jako samostatnou aulu, kde uživatelé budou moci vytvářet a využívat myšlenkové paláce. K tomu využijte následující postup:

1. Seznamte se podrobně s odbornou literaturou o metodě loci a proveďte její rešerši.
2. Prostudujte a naučte se pracovat v systému virtuální reality Unreal.
3. Navrhněte základní organizaci a strukturu 3D scén pro implementaci metody loci.
4. Navrhněte a vytvořte 3D objekty těchto scén.
5. Vytvořte a zkompletujte navržené scény v systému Unreal.
6. Ověřte implementaci metody loci v praktických podmínkách.
7. Optimalizujte navrženou implementaci.
8. Pokuste se stanovit efektivitu virtuální implementace metody loci a porovnejte ji s nevirtuální variantou.

Bakalářská práce

METODA LOCI VE VIRTUÁLNÍ REALITĚ

Dominik Pupala

Fakulta informačních technologií
Katedra softwarového inženýrství
Vedoucí: doc. Ing. Mgr. Petr Klán, CSc.
12. května 2022

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2022 Dominik Pupala. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení, je nezbytný souhlas autora.

Odkaz na tuto práci: Pupala Dominik. *Metoda loci ve virtuální realitě*. Bakalářská práce. České vysoké učení technické v Praze, Fakulta informačních technologií, 2022.

Obsah

Poděkování	vi
Prohlášení	vii
Abstrakt	viii
Seznam zkratek	ix
Úvod	1
1 Metoda loci	3
1.1 Použití	3
1.2 Interaktivní implementace	4
1.2.1 Implementace jako experimentální prostředí	4
1.2.2 Implementace a zobrazovací zařízení	5
1.2.3 Implementace jako teoretický koncept	6
2 Unreal Engine	7
2.1 Vývoj pro VR	7
2.2 Programování a skriptování	8
2.2.1 C++	8
2.2.2 Blueprinty	8
2.3 Herní framework	9
2.4 Nástroje	10
3 Návrh řešení	11
3.1 Interpretace metody loci	11
3.1.1 Prostředí	11
3.1.2 Scény	12
3.1.3 Objekty	13
3.1.4 Ostatní	13
3.2 Funkční a nefunkční požadavky	13
3.2.1 Obecné	13
3.2.2 Metoda loci	14
3.3 Výběr technologií	15
3.4 Návrh základních tříd	16

3.4.1	Třída Item	16
3.4.2	Třída Furniture	17
3.4.3	Třída PlayerCharacter	18
4	Implementace řešení	19
4.1	Prostředí	19
4.1.1	Místnost	19
4.1.2	Objekty scén	20
4.1.3	Logika	21
4.2	Intro	23
5	Test řešení	25
5.1	Memorizace množiny objektů	25
5.1.1	Virtuální metoda loci	25
5.1.2	Tradiční metoda loci	26
5.2	Memorizace posloupnosti objektů	28
5.2.1	Virtuální metoda loci	28
5.2.2	Tradiční metoda loci	29
5.3	Výsledky	30
	Závěr	33
	Obsah přiloženého média	37

Seznam obrázků

2.1	Ukázka blueprintu – aktualizování komponent postavy hráče	9
3.1	Diagram tříd popisující položky k zapamatování/testování	17
3.2	Diagram tříd popisující nábytek a loci	18
4.1	Ukázka z tvorby úrovně reprezentující metodu loci	20
4.2	3D objekty reprezentující položky k zapamatování/testování a nábytek	21
4.3	Logika delegace režie v blueprintu ApplicationManager	22
4.4	Použití komponent v blueprintu PlayerCharacter	22
4.5	Logika přepnutí úrovně v blueprintu IntroCharacter	23
5.1	Rozložení nábytku ve virtuální místnosti	26
5.2	Vyhodnocení kvízové fáze aplikace	27
5.3	Náčrt ložnice s nábytkem	27
5.4	Rozložení nábytku a loci ve virtuální místnosti (včetně pořadí)	28
5.5	Postupné ukládání objektů do loci ve virtuální místnosti	29

Seznam tabulek

5.1	Shrnutí dob trvání obou pokusů (v minutách)	31
5.2	Porovnání vlastností variant metody loci	31

Chtěl bych poděkovat především doc. Ing. Mgr. Petru Klánovi, CSc. za vedení této práce, jeho cenné rady a zkušenosti. Dále mé díky patří mým nejbližším přátelům, kteří mě podporovali při studiu.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací. Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 citovaného zákona.

V Praze dne 12. května 2022

.....

Abstrakt

Předmětem této bakalářské práce je návrh a implementace metody loci ve virtuální realitě. První část práce se zabývá problematikou zmiňované mnemotechnické pomůcky, analýzou jejích interaktivních implementací a vývojem v systému Unreal Engine 4. Druhá část práce představuje návrh a implementaci aplikace pro zařízení HTC Vive. Nakonec je implementace porovnána s tradiční podobou metody loci.

Klíčová slova metoda loci, mnemotechnická pomůcka, virtuální realita, interaktivní desktopová aplikace, Unreal Engine, blueprints, HTC Vive

Abstract

The purpose of this bachelor thesis is the design and implementation of the method of loci as a virtual reality application. The first part of this thesis deals with the topic of beforementioned mnemonic device, including the analysis of already existing interactive implementations, and the development inside the Unreal Engine 4. The second part presents design and implementation process of the application for the goal device, HTC Vive. Finally, the application is compared to the traditional form of the method of loci.

Keywords method of loci, mnemonic device, virtual reality, interactive desktop application, Unreal Engine, blueprints, HTC Vive

Seznam zkratk

VR	Virtuální realita
AR	Rozšířená realita
UE4	Unreal Engine 4
UHT	UnrealHeaderTool
3D	Trojrozměrný
2D	Dvourozměrný
UI	Uživatelské rozhraní
AI	Umělá inteligence

Úvod

Virtuální realita (VR) se stává čím dál dostupnější pro široké publikum, a to zejména v oblasti zábavy. VR technologie však nachází svůj užitek i v profesionálních sférách, jako je například lékařství či průmysl, a to především díky imerzivní interakci, které není jednoduché dosáhnout na klasickém zobrazovacím zařízení. Akademici, technici i umělci tak mají nový nástroj k činnosti ve svých odvětvích. Jedním ze zajímavých témat v odvětví kognitivní psychologie, disciplíny psychologie zabývající se tématy jako zpracovávání informací, pozornosti, vnímání či paměti, je metoda loci.

Metoda loci, známa také jako myšlenkový palác nebo paměťová cesta, je memorizační metoda a jedna z mnemotechnických pomůcek, která byla využívána řečníky již za dob starověkého Řecka a Říma. Její princip spočívá ve vizualizaci známého prostředí a jeho jednotlivých míst (loci), která slouží k následné asociaci s položkami k zapamatování. Jinými slovy si uživatel této metody představí jemu známé prostředí a rozdělí si jej na dílčí loci, do kterých při imaginárním průchodu ukládá položky, které si chce zapamatovat. Pro vzpomnutí na uložené položky si uživatel prostředí znova představí a začne jím pomyslně procházet.

Tato bakalářská práce pojednává o návrhu a implementaci metody loci ve VR, konkrétně v systému Unreal Engine 4 (dále jen UE4) pro zařízení HTC Vive. Na základě analýzy je navrhována interpretace zmiňované memorizační metody, jejíž podoba je rozdělena do dílčích scén obsahující různé objekty, které odehrávají specifické role vyplývající z navržených interaktivních mechanismů. Tento návrh je posléze implementován a porovnán s tradiční podobou metody loci. Je nutné podotknout, že porovnání není provedeno z hlediska kontrolních skupin, nýbrž z hlediska autora práce a poslouží ke stanovení potenciálních optimalizací interpretace metody.

Motivací autora je prozkoumat problematiku a úskalí interaktivní reprezentace memorizační metody, kterou je kvůli individualitě vnitřních mentálních procesů obtížné zkoumat v její tradiční podobě, a také potenciálně představit metodu loci v interaktivní podobě širšímu publiku. K motivaci také přispěla účast autora na konferenci Czech VR Fest v roce 2019, kde přednášející z NUDZ (Národní ústav duševního zdraví) seznámili posluchače s projekty uplatňující terapeutické využití VR pro rehabilitaci oslabených kognitivních funkcí a pro expoziční terapii.

Hlavním cílem této práce je tedy navrhnout a implementovat vlastní řešení představující metodu loci jako VR aplikaci v UE4. V teoretické části práce je cílem představit metodu loci

a analyzovat současná řešení jejích interaktivních implementací. Dalším dílčím cílem teoretické části je seznámit se s vývojem VR aplikací v UE4. V praktické části práce je cílem navrhnout vhodnou interpretaci metody loci a organizaci souvisejících objektů. Organizací objektů se myslí jejich třídní hierarchie, 3D reprezentace a možné interakce mezi sebou. Posledním cílem práce je implementovat výsledné návrhy v kompletní VR aplikaci a tuto aplikaci následně porovnat s tradiční podobou metody loci.

Metoda loci

Metoda loci je memorizační metoda, která je jednou z nejstarších a neúčinnějších mnemotechnických pomůcek z dob starověkého Řecka [1]. Mezi mnemotechnické pomůcky patří mj. i užití klíčových slov, akronymů či akrostik, tj. techniky využívající již známých informací k zapamatování nových informací [2].

Metoda loci, známá také jako myšlenkový palác či paměťová cesta, využívá znalosti prostředí, které je rozděleno na dílčí klíčová místa (loci), ty slouží k asociaci s novými informacemi. Loci jsou typicky uspořádána v nějakou posloupnost, díky níž je možné memorizovat informace v určitém pořadí. Protože prostředí i informace k zapamatování mohou mít téměř jakoukoliv podobu, je tato mnemotechnická pomůcka velice flexibilní. Prostředím může být cokoliv, v čem se lze prostorově orientovat, typicky se jedná o místnosti, budovy či trasy. Informace k zapamatování mohou být (mimo typické hmatatelné objekty) i koncepty jako např. barvy, vzpomínky či vnořené myšlenkové paláce. Limitujícím faktorem je individuální schopnost asociovat danou informaci s loci. [3, 1]

1.1 Použití

Typická sekvence použití metody loci začíná výběrem prostředí, které uživatel metody dobře zná, například obývacího pokoje či cesty do práce. V tomto prostředí si uživatel stanoví klíčová loci. V obývacím pokoji by taková loci tvořily např. kusy nábytku či obrazy, zatímco na cestě do práce by mohlo jít o přístupné zastávky, sochy či výlohy kamenných obchodů. Uživatel si vytvoří mentální podobu vybraného prostředí a začne jim postupně procházet kolem klíčových loci, ke kterým asociuje informaci k zapamatování. Pro vzpomnutí si na asociované informace se uživatel znovu začne procházet v mentální podobě prostředí. [3, 1]

Takto lze použít metodu loci k zapamatování libovolných souborů informací, od nákupních seznamů přes osnovy mluveného projevu až k usazení hostů na svatební hostině. Pro další soubory informací není nutné využívat nového prostředí – je možné znovu zužitkovat již použité prostředí, a to dokonce beze ztráty na efektivitě. [1]

Metoda loci je také používána v memorizačních soutěžích a podle Maguirové je to nejpou-

žívanější nástroj mezi předními paměťovými atlety, kde pouze každý desátý atlet uvádí použití jiného nástroje [4]. Používány jsou i vlastní upravené varianty metody loci, např. systém paměťového města pro jazyky, který představil osminásobný světový šampión v memorizaci Dominic O'Brien [5].

1.2 Interaktivní implementace

Metoda loci a její nevysvětlitelná efektivita je jednou ze záhad kognitivní psychologie, na kterou kognitivní vědci stále hledají vysvětlení. Přestože se jedná o velice starou mnemotechnickou pomůcku, je počet publikovaných studií velice skromný. Výzkum metody loci komplikuje nejen její závislost na individuálních vnitřních mentálních procesech, ale i nutnost rozsáhlého tréninku účastníků potenciální studie. Příchod moderních technologií nabídl výzkumníkům příležitost, jak tyto problémy potenciálně vyřešit a zkoumat metodu loci dále, a to ve formě interaktivních nástrojů. [1, 6]

Během rešerše byly nalezeny tři články zabývající se interaktivní implementací metody loci. Dva články popisují použití interaktivních implementací ve svých studiích, zatímco zbylý článek popisuje, jak by taková interaktivní implementace metody loci měla vypadat.

1.2.1 Implementace jako experimentální prostředí

Legge ve své studii představuje interaktivní implementaci jako řešení lépe kontrolovatelného prostředí pro výzkum. Samotná studie pojednává o porovnání tradiční a modifikované podoby metody loci, a to z hlediska efektivity při memorizaci deseti rozdílných seznamů o jedenácti slovech. [1]

Modifikovaná podoba metody loci se od té tradiční liší pouze v úvodní fázi, tedy v utváření prostředí a jeho členění na loci. Namísto volné volby známého prostředí je skrze obrazovku počítače¹ představeno virtuální prostředí. [1]

V rámci studie byli účastníci rozděleni do tří skupin. První skupina byla instruovaná k použití tradiční podoby metody loci, tedy použití vlastního prostředí, zatímco druhá skupina využívala pro metodu loci virtuální prostředí. Zbylá kontrolní skupina nebyla instruovaná k použití žádné konkrétní metody či mnemotechnické pomůcky. [1]

Použita byla tři různá virtuální prostředí, která byla namodelována podle skutečných předloh (byt, škola a sklad), a jedno úvodní, které sloužilo k instruktáži navigace po prostředí. Účastníci všech tří skupin byli nejdříve vyškoleni v úvodním prostředí a poté jim bylo představeno jedno ze zbylých tří, se kterým se měli během pěti minut seznámit. [1]

Výsledek studie neprokázal žádný zásadní rozdíl v efektivitě mezi tradiční a virtuální variantou metody loci, a potvrdil tak hypotézu, že předložením předem připraveného prostředí ve virtuální podobě lze metodu loci využít beze ztráty na efektivitě. [1]

¹iMac 5.1 s monitorem o úhlopříčce 15"

Diskuze o článku

Studie a její nečekaný výsledek – využití virtuálního prostředí beze ztráty na efektivitě – implikuje pouze to, že pro použití metody loci není potřeba skutečného prostředí. Ačkoliv jsou účastníci této studie vystaveni virtuálnímu prostředí jen na krátkou dobu (5 minut), lze argumentovat, že nějaký rozsah familiarizace s prostředím probíhá, a to alespoň v podobě možnosti se v něm volně pohybovat.

Zvolená realizace všech tří virtuálních prostředí uživatelům nevnučuje způsob členění loci ani jak k nim asociovat informace k zapamatování. Jedná se samozřejmě o důsledek skutečnosti, že virtuální prostředí pouze nahrazuje úvodní proces výběru reálného známého prostředí a nemá žádnou další roli v metodě loci.

Autoři publikace však skutečně přišli se způsobem, jak vytvořit lépe kontrolovatelné experimentální prostředí. Tento způsob může být využit v dalších studiích mentálních procesů metody loci.

1.2.2 Implementace a zobrazovací zařízení

Krokos, Plaisant a Varshney ve své studii používají dvě interaktivní implementace, jednu využívající VR zařízení² a druhou obrazovku počítače³. Studie porovnává obě implementace z hlediska efektivity při memorizaci obličejů. [7]

Účastníci studie byli nejdříve seznámeni se 42 obličejí známých postav a jejich jmény. Poté byli rozděleni do dvou skupin, přičemž jedna začínala v implementaci pro VR zařízení, zatímco druhá v implementaci pro typické zobrazovací zařízení. Po dokončení experimentu v první implementaci se účastníci přesunuli do implementace druhé, ve které experiment opakovali. [7]

Implementace se lišily pouze v zobrazovacím zařízení a v použitém prostředí. Jednalo se o prostředí představující palác a středověké město. V těchto prostředích bylo již předem ručně vloženo 21 obličejů, tedy 42 v obou prostředích dohromady. Uživatelé měli ve středu těchto prostředí statickou pozici a mohli se pouze otáčet. [7]

Experiment byl členěn na fáze, kde v první fázi byly vidět obličeje a ve druhé se zobrazovaly čísla pozic, ke kterým uživatelé měli přidělit jméno osoby na základě zapamatovaného obličeje. Přidělení jména k obličejí probíhalo verbálně a bylo zaznamenáváno dohlížejícím výzkumníkem. [7]

Výsledek studie prokázal statisticky zásadní rozdíl mezi interaktivními implementacemi, a to ve prospěch implementace využívající VR jako zobrazovací zařízení. Autoři článku také narazili na efekt pořadí, kde se výsledky účastníků zlepšily, když se z varianty s typickým zobrazovacím zařízením přesunuli na variantu ve VR. Naopak se výsledky účastníku zhoršily, pokud začali variantou ve VR. [7]

²Oculus DK2 HMD

³monitor o úhlopříčce 30"

Diskuze o článku

Ačkoliv se článek opírá o publikace zabývající se metodou loci a implementace jsou uchopeny jako „virtuální paláce“ [7], není na principy metody loci brán žádný ohled. To je potvrzeno skutečností, že účastníci studie nejsou v žádném bodě seznámeni s touto mnemotechnickou pomůckou.

Interaktivní implementace v tomto článku jsou spíše návodem, jak by metoda loci neměla být implementována. Ve virtuálních prostředích jsou předvybrané nejen jednotlivá loci, ale i způsob asociace k nim či samotná informace k zapamatování. Uživatelé tedy, kromě rozhlížení se kolem sebe, nemohou s prostředím nijak interagovat.

1.2.3 Implementace jako teoretický koncept

Peeters a Segundo-Ortin ve své publikaci pojednávají o problematice dosavadních interaktivních implementací metody loci a přehodnocením současného přístupu kognitivní psychologie představují návrhy a doporučení pro budoucí interaktivní implementace, především pro aplikace ve VR. [6]

Je kladen důraz na důležitost familiarizace uživatele s prostředím, které má být dostatečně velké a imerzivní. Uživatel by neměl být omezován v pohybu po prostředí a samotné prostředí by mělo být přístupné ve všech fázích metody loci. [6]

Ukládání informací do jednotlivých loci by mělo probíhat ve formě obrázků. Nejsou však zamýšleny přímé obrázky, ale snímky sestavené uživatelem, jež by zachycovaly libovolně velikou sadu objektů v zátiší. [6]

Objekty ve snímku by měly být součástí velké databáze, se kterou může uživatel v reálném čase interagovat. Objekty tvořené snímky, či dokonce jimi naplněné prostředí, by mělo být možné sdílet s dalšími uživateli. [6]

Diskuze o článku

Teoretický rámec v tomto článku představuje velice dobrý základ pro podobu interaktivní implementace metody loci, která zachycuje všechny její dílčí části – prostředí je možné používat ve všech fázích metody. Taková implementace má potenciál být lépe přístupná širokému publiku jako ucelená aplikace.

Reprezentace prostředí a členění na loci nejsou striktní, pouze způsob asociace loci s novou informací je vnučen (vkládání obrázků/snímku). V prostředí je dbáno na možnost uživatele volně se pohybovat.

Snímky sestavené z 3D objektů jsou však zajímavou myšlenkou. Řeší totiž abstraktní koncepty (např. barvy), a to tak, že si uživatel může sám zvolit sadu objektů, které si sestaví do zátiší. Uživatel si tak tvoří vlastní zakódování informace.

Možnost sdílet prostředí a snímky mezi uživateli je spíše funkcionalita pro potenciální aplikaci, ale nemá žádný vliv na efektivitu samotné metody loci.

Unreal Engine

Unreal Engine je populární herní engine vyvíjený společností Epic Games, která jej používá mj. i pro svůj přední titul Fortnite [8]. Tento herní engine je však oblíbený i mezi ostatními velkými herními studii, např. studio CD Projekt ohlásilo přechod z vlastního enginu na Unreal Engine 5¹ pro nového Zakláče [9]. Popularitě tohoto nástroje zajisté napomáhá i jeho přístupnost pro nezávislé vývojáře či malá studia, a to především díky platebnímu modelu. Epic Games umožňuje bezplatné využití, jestliže roční zisk nečiní více než milion amerických dolarů a případně požaduje 5 % z kvartálního obrátu [10]. Těchto 5 % je prominuto, je-li aplikace publikována na jejich platformě Epic Store [11].

V roce 1995 započal Tim Sweeney, zakladatel a výkonný ředitel Epic Games, vývoj Unreal Engine, který sloužil k produkci hry Unreal. Přestože byl engine zpočátku navrhnut pro tvorbu videoher (konkrétně stříleček z první osoby), je v současné době ve velké míře využíván i v jiných průmyslech, jako je např. film, architektura či automotive. [11]

Ačkoliv engine není open-source, je dostupný pro všechny registrované uživatele v repozitáři na platformě GitHub [11]. Vývojáři tak mohou podle svých potřeb upravovat chyby a vyvíjet nové funkcionality, případnými změnami je dokonce možné přispět do vývoje enginu [12].

2.1 Vývoj pro VR

Unreal Engine podporuje většinu VR a AR platform [13]. Přední VR platformy jako SteamVR, Oculus VR či Windows Mixed Reality jsou samozřejmostí. Pro každou cílovou platformu je potřeba splnit externí podmínky (např. pro SteamVR je potřeba mít nainstalovanou aplikaci Steam [14]), ale jinak se vývoj VR aplikací v Unreal Engine nijak výrazně neliší oproti vývoji pro typická zobrazovací zařízení. K dispozici jsou připravené komponenty a případně VR šablona pro nový projekt, obsahující základní třídy a objekty. Vývojář, který už nějaké zkušenosti s tvorbou v enginu má, by tedy neměl mít žádné problémy s vytvořením své první aplikace pro VR.

¹představen v dubnu 2022

2.2 Programování a skriptování

Programování v Unreal Engine je možné rozdělit do tří úrovní [15]:

1. programování enginu
2. programování aplikace
3. skriptování

Programování enginu je řešení nízkourovňových úloh, např. tvorba nástrojů či technologií, které slouží k vývoji aplikace, ale zároveň mají obecný rámec užití i mimo specifický projekt [15]. Jedná se o modifikace zdrojového kódu samotného a je pochopitelně použit stejný jazyk, ve kterém je Unreal Engine vyvíjen – C++ [11].

Programování aplikace a skriptování představují řešení vysokoúrovňových úloh, které tvoří logiku konkrétní aplikace. Hranice mezi programováním a skriptováním je z hlediska vývoje aplikace velice volná, protože za skriptování lze považovat cokoli, co nezasahuje do kódu enginu. Programováním logiky aplikace se nicméně myslí zpravidla implementace fundamentálních systémů a struktur, které tvoří jádro aplikace. Skriptováním se pak myslí manipulace s konkrétními objekty v rámci zmiňovaných systémů a struktur [15]. Nezávisle na tom, zdali se jedná o programování či skriptování, je možné tvořit logiku aplikace v C++ nebo v blueprintech.

Rozhraní enginu je pro C++ i blueprints stejné – v obou přístupech se systémy enginu používají téměř totožným způsobem [15]. Oba dva zmiňované způsoby tvoření logiky aplikace je možné kombinovat a jedná se dokonce o doporučený přístup pro vývoj v Unreal Engine [16].

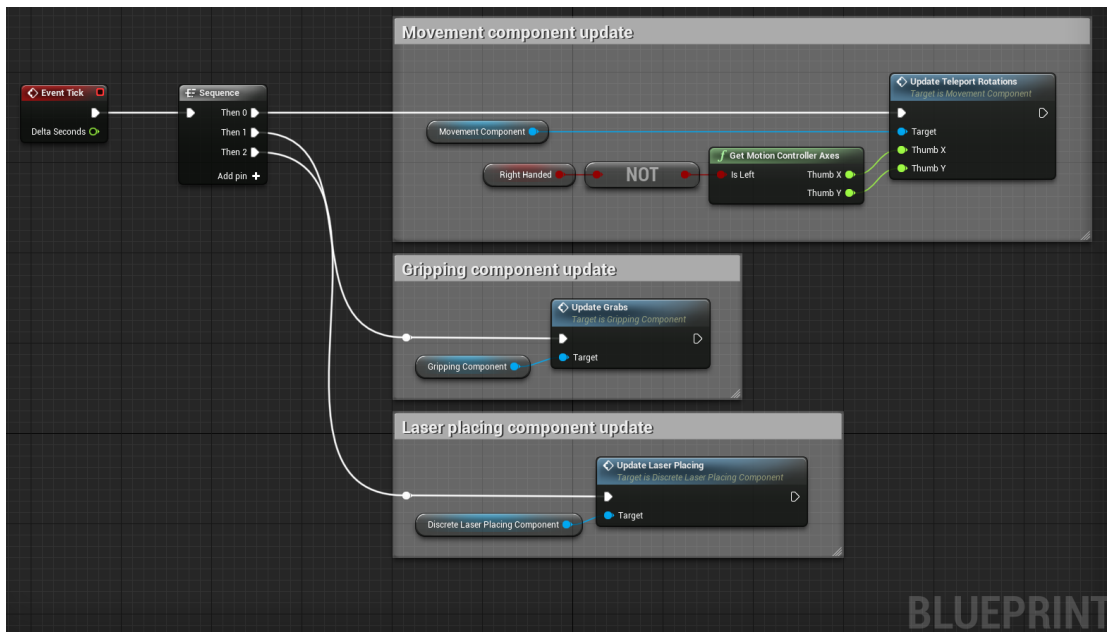
2.2.1 C++

V C++ je možné řešit vysokoúrovňové úlohy díky anotacím pro UnrealHeaderTool (dále jako UHT), což je nástroj pro generování a čtení kódu podporující systém herních objektů (tzv. UObject). UHT je spuštěn před nativním kompilátorem, aby prohledal hlavičkové C++ soubory, ve kterých vyhledává metadata enginu. Ty slouží k automatickému vygenerování kódu implementujícím různé funkcionality UObject systému, jako je např. zmiňovaná reflexe. [17, 18]

Jednou z největších nevýhod C++ (jakožto řešení vysokoúrovňových úloh) je skutečnost, že je nutné každou změnu zkompileovat. S provedenými změnami se totiž bude kompilovat i engine samotný – což je velice časově náročné. Vývojové prostředí navíc není součástí enginu a je také nutné použít nástroj třetí strany. Výhodou C++ je jeho výkon, který může být zásadní v kritických systémech velkých aplikací [15].

2.2.2 Blueprints

Blueprint je objekt enginu (rozšiřuje nějaký objekt z UObject systému), ve kterém je pomocí vizuálního uzlového prostředí možné vytvářet logiku aplikace bez znalosti syntaxe jazyka C++. Ve zmiňovaném prostředí jsou jednotlivé funkce reprezentovány jako samostatné uzly. Propojením těchto uzlů je pak tvořena komplexní logika (viz obrázek 2.1). [19]



■ **Obrázek 2.1** Ukázka blueprintu – aktualizování komponent postavy hráče

Výhodou blueprintů je jejich rychlost a přístupnost při vytváření prototypů. Bluepriny nevyžadují pokročilé znalosti programování a umožňují i ostatním členům týmu psát své skripty nezávisle na programátorovi. Nevýhodou blueprintů je nepraktičnost při verzování – typické verzovací systémy s nimi neumí pracovat jako s kódem. Každá změna navíc vyžaduje uložení, a to i když se jedná pouze o vizuální úpravu (např. posunutí uzlu funkce).

2.3 Herní framework

Herní framework (tzv. Gameplay Framework) je kolekce systémů a tříd reprezentující herní pravidla, uživatelské rozhraní atd. Framework v Unreal Engine stanovuje nejen průběh aplikace, určuje i základní organizaci pro herní objekty. Ačkoliv je terminologie založena na herním vývoji, je framework využíván pro většinu aplikací i mimo herní průmysl. [20]

Průběh aplikace řídí systém založený na třídách GameMode, GameState a PlayerState. GameMode třída existuje pouze na serveru (na vzdáleném nebo lokálním) a je definicí aplikace – obsahuje všechny ostatní třídy herního frameworku. GameState třída slouží k udržování stavu aplikace, jako např. seznam hráčů a jejich skóre. Stav hráče či uživatele je pak udržován ve třídě PlayerState. Obě třídy udržující stav se nachází lokálně na počítačích jednotlivých uživatelů, kde GameState třída existuje pouze jako kopie ze serveru. [20]

Třídy Pawn a Character slouží k reprezentaci agentů ve scéně. Ačkoliv se agentem obvykle myslí nějaká bytost či postava, může jim být cokoli, co je kontrolováno uživatelem nebo AI. Třída Pawn je nejprostší forma agenta a umožňuje přijímat uživatelský vstup (nebo vstup od AI). Character je pak rozšířením předešlé zmiňované třídy, jejíž součástí jsou funkcionality relevantní pro humanoidní postavy, např. pohyb a kolize. [20]

Agenti jsou kontrolováni přes rozhraní tříd tzv. Controllers. Tato rozhraní zpracovávají uživatelský vstup ze vstupních zařízení (klávesnice, myš atd.) nebo je generují na základě AI logiky (behaviorální stromy). Třída Controller je nejzákladnější forma takového rozhraní a dědí z něj třídy PlayerController a AIController, jejichž rozdíl je vysvětlen v předchozí větě. [20]

Pro zobrazování informací uživatelům slouží třídy HUD a PlayerCameraManager. HUD vykresluje veškeré UI prvky pro 2D zobrazovací zařízení, zatímco PlayerCameraManager představuje pohled uživatele v 3D prostoru a mj. umožňuje i aplikovat efekty na jednotlivé kamery (např. rozmazání). [20]

2.4 Nástroje

Součástí Unreal Engine je rozsáhlá sada nástrojů, editorů a systémů. Nástroj je funkcionalita, která řeší specifickou úlohu. Editor je tvořen kolekcí nástrojů a slouží k řešení komplexních úloh. Systémem se myslí sada funkcionalit, které dohromady slouží k produkci konkrétního aspektu aplikace (např. animace, UI či AI). [21]

Mezi základní editory patří Level Editor a Blueprint Editor. Level Editor je implicitní editor, který je uživateli zobrazen po spuštění engine, a slouží k tvorbě úrovní/scén ve hře/aplikaci. Blueprint Editor slouží k tvorbě blueprintů a používá se tedy při vývoji všech aplikací, které svoji logiku neimplementují čistě v C++. [21]

Využití nástrojů a editorů se liší nejen dle rámce aplikace, ale také pozicí či rolí konkrétního vývojáře. Designéři pro práci s 3D objekty používají Static Mesh Editor a Material Editor. Static Mesh Editor slouží k úpravě 3D objektů a jejich vlastností (kolize, úroveň detailu atd.) pro engine, zatímco Material Editor slouží k tvorbě materiálů, které určují vizuální podobu zmiňovaných objektů. UMG UI Editor se využívá při návrhu uživatelského rozhraní aplikace. [21]

K animaci 3D objektů poslouží animátorům Persona Editor, zatímco zvukaři ocení nástroje v editoru Sound Cue Editor, ve kterém mohou upravovat a míchat zvukové efekty. Pro skriptování NPC (nehratelných postav) využijí AI vývojáři Behavior Tree Editor. [21]

Návrh řešení

Tato kapitola popisuje návrh interaktivní implementace metody loci pro VR zařízení – konkrétně HTC Vive. Na začátku je popsána interpretace metody loci vycházející ze zadání a z analýzy kapitoly 1.2. Interpretace poté slouží jako základ pro funkční a nefunkční požadavky aplikace. Na konci kapitoly je představen návrh základních tříd.

3.1 Interpretace metody loci

Metoda loci je velice flexibilní a implementace v její plné koncepci je takřka nemožná. Interpretace metody loci proto slouží ke specifikaci, která konkrétní metodika bude vybrána pro návrh aplikace, jakým způsobem bude metodika v aplikaci reprezentována, jaké objekty bude možné memorizovat a mnoho dalšího.

3.1.1 Prostředí

Pro interpretaci metody loci je zvolena varianta představující prostředí jako prostornou místnost, ve které jsou loci reprezentovány nábytkem. Použití místnosti je jedinou používanou variantou pro interaktivní implementace (viz kapitola 1.2), a to i přesto, že je možné prostředí metody loci reprezentovat i jinými koncepty využívající prostorových vztahů mezi loci (viz kapitola 1.1). Nejspíše se jedná o důsledek vycházející z praktičnosti implementování jednotlivých variant, nicméně v této práci je tak učiněno ze zadání, které specifikuje použití auly – velká místnost.

Význam slova „aula“ představuje spíše slavnostní síň či sál, ale vzhledem k důležitosti familiarizace s prostředím by měla být pro návrh aplikace zvolena místnost, která bude mít podobu typického každodenního prostředí. Jedná se spíše o stylistické rozhodnutí, které plyne z estetiky a podoby nábytků pro daná prostředí – méně pravidelné a více osobité je zde považováno za klíčový faktor pro familiarizaci. Místnost musí být dostatečně prostorná, aby v ní bylo možné rozložit nábytek, který dohromady bude tvořit deset pozic pro loci.

3.1.2 Scény

Většina interaktivních implementací metody loci je využívána jen jako nástroj nahrazující prostředí. Přitom se jedná pouze o malou část této mnemotechnické pomůcky, a proto je důležité, aby interpretace metody loci zahrnovala všechny její kroky. Pro návrh aplikace je tedy použito následující členění do scén:

1. familiarizační fáze
2. memorizační fáze
3. kvízová fáze

Tyto scény se nebudou lišit fyzickou polohou ani podobou prostředí (vyjma rozložení nábytku a memorizačních objektů), nýbrž možnými akcemi, které uživatel v určité scéně smí provést.

Mezi scénami by mělo být uživateli umožněno přepínat, a to tak, aby nedocházelo k chybám ve formě tzv. softlocků, které sice nezapříčiní pád programu, ale zároveň neumožní další postup v aplikaci.

Familiarizační fáze

Familiarizační fáze v obecné metodě loci představuje seznámení s prostředím a selekci loci k asociaci pro případnou memorizaci. Pro návrh aplikace je toto koncipováno za pomoci rozložení nábytku. Jak již bylo řečeno, nábytek představuje loci, ale také utváří strukturu místnosti, a umožňuje tedy uživateli tvorbu vlastního prostředí – skrytá familiarizace.

Pokládání nábytku je pro tuto scénu stěžejní a mělo by být jak intuitivní, tak i snadno replikovatelné – mělo by proto využívat diskrétní transformace v podobě statických kroků a pravoúhlých rotací. Nábytek nesmí být možné pokládat na již obsazenou pozici.

Memorizační fáze

Memorizační fáze slouží k zapamatování objektů, a to tak, že si uživatel asociuje objekty k zapamatování s loci v prostředí. Toto by mělo být realizováno generováním objektů k memorizaci do scény, které uživatel bude moci ukládat do jednotlivých loci – uživateli tedy musí být také umožněno objekty volně přenášet po prostředí.

Objektů se musí vygenerovat stejný počet jako loci ve scéně a nesmí být umožněno uživateli uložit dva objekty do stejného loci.

Kvízová fáze

K ověření úspěšné memorizace objektů slouží kvízová fáze. Ta je pro návrh aplikace koncipována jako testovací minihra, která ve všech loci nejprve skryje uložené objekty k memorizaci, načež začne generovat nové objekty do scény. Tyto objekty jsou generovány ze stejné množiny jako objekty v memorizační fázi, jelikož je uživatel bude muset vkládat do příslušných loci, aby si ověřil, že si je zapamatoval správně.

V této scéně by se tedy měly skrýt zapamatované objekty v loci, začít generovat testovací objekty a na konci scény by mělo proběhnout vyhodnocení, které v jednotlivých loci porovná zapamatované objekty s testovacími. Vyhodnocení by také mělo rozlišovat objekty nepatřící do prostředí od objektů ve špatném loci.

3.1.3 Objekty

Jelikož abstraktní koncepty nelze ve 3D prostředí jednoduše reprezentovat jako samostatné objekty, jsou pro memorizaci a testování v interpretaci metody loci zvoleny hmatatelné objekty. Generování objektů k memorizaci a testování musí být náhodné a zároveň pocházet z množiny složené z alespoň patnácti položek.

Objekty ve hře by měly být navrženy tak, aby bylo možné snadno přidávat jejich další varianty, a to především položky reprezentující nábytek a objekty k zapamatování/testování.

3.1.4 Ostatní

Vzhledem k polemizovaným důsledkům motoriky na efektivitu familiarizace a kognitivních procesů je důležité, aby v žádné scéně nebyl uživatel omezován v pohybu. Návrh aplikace by mimo obvyklou teleportaci měl zvažovat i jiné způsoby pohybu, především varianty s plynulým optickým tokem při změně pozice – jinými slovy způsoby pohybu, které simulují plynulou změnu v prostoru beze skoku.

Jestliže návrh aplikace neumožňuje obojetné ovládní, musí být všechny vstupní mechanismy navrženy jak pro praváky, tak i pro leváky. Případný výběr režimu pro praváky či leváky musí být součástí aplikace a nejlépe by se mělo uživateli nabídnout ihned po spuštění.

Pro zpestření aplikace by součástí interpretace metody loci měl být další testovací režim, který v kvízové fázi nejen náhodně vygeneruje objekty, ale přímo je mezi ně rozdělí – uživatel tedy bude muset objekty přemístit do správných loci, vyloučit objekty, které nepatří do memorizovaného listu položek, a nahradit je těmi správnými.

3.2 Funkční a nefunkční požadavky

Navržená interpretace metody loci sloužila jako náhrada za zadávací dokumentaci a na jejím základě byly sepsány funkční a nefunkční požadavky.

3.2.1 Obecné

Funkční požadavky

- **FA01: Uživatel se bude moci po prostředí pohybovat**

Ve všech fázích metody loci bude uživateli umožněn pohyb teleportováním a alespoň jedním dalším způsobem, který nemění pozici skokem.

- **FA02: Všechny interaktivní mechanismy budou obojetné**

Veškeré interaktivní mechanismy, u kterých nelze použít totožné schéma pro ovladače, budou mít možnost pro změnu dominantní ruky. Případná konfigurace tohoto nastavení bude umožněna po spuštění aplikace.

Nefunkční požadavky

- **NA01: Aplikace bude vyvinuta v UE4**

Aplikace bude vyvinuta v nejaktuálnější verzi čtvrté generace systému Unreal Engine (verze 4.27).

- **NA02: Aplikace bude cílená pro zařízení HTC Vive**

Aplikace bude plně kompatibilní s VR zařízením HTC Vive (nativní platforma SteamVR).

3.2.2 Metoda loci

Funkční požadavky

- **FAM01: Aplikace bude zahrnovat familiarizační fázi metody loci**

Součástí aplikace bude scéna reprezentující familiarizační fázi metody loci. V této scéně bude uživatel moci rozkládat loci (v podobě nábytku) po prostředí, a to v diskretní mřížce (posuny o statické kroky a rotace v pravých úhlech). Mezi jednotlivými kusy nábytku nesmí docházet k překryvu.

- **FAM02: Aplikace bude zahrnovat memorizační fázi metody loci**

Součástí aplikace bude scéna reprezentující memorizační fázi metody loci. Tato scéna bude generovat do prostředí položky k zapamatování, které uživatel bude moci přemísťovat a vkládat do jednotlivých loci. Scéna bude generovat přesně 10 položek k zapamatování.

- **FAM03: Aplikace bude zahrnovat kvízovou fázi metody loci**

Součástí aplikace bude scéna reprezentující kvízovou fázi metody loci. Tato scéna bude generovat do prostředí položky k testování, poté co skryje veškeré položky k zapamatování ve všech loci. Do vyprázdněných loci bude uživatel vkládat zmiňované položky k testování. Scéna bude postupně generovat položky z množiny všech položek k zapamatování/testování.

- **FAM04: Aplikace bude vyhodnocovat kvízovou fázi metody loci**

Na konci scény reprezentující kvízovou fázi metody loci bude pro uživatele vyhodnoceno, jak si při testu vedl. Vyhodnocení bude rozlišovat položky nepatřící do množiny položek k zapamatování, položky patřící do množiny, avšak umístěné do špatných loci a správně přiřazené položky.

- **FAM05: Aplikace bude mít alternativní možnost pro kvízovou fázi metody loci**

Na počátku aplikace bude možné vybrat alternativní průběh scény reprezentující kvízovou fázi metody loci. Alternativní varianta scény bude rozšiřovat tu původní o náhodné rozmístění položek k testování přímo do loci. Uživatel tyto položky bude muset přemístit do správných loci.

- **FAM06: Uživatel bude moci mezi jednotlivými scénami přepínat**

Uživatel si bude moci sám řídit postup mezi scénami, avšak pouze za předpokladu, že bude plnit podmínky zabraňující chybám způsobující nemožnost dalšího postupu v aplikaci. Uživatel tedy bude muset pro postup do scény reprezentující memorizační fázi vložit do prostředí všechny nábytek, zatímco pro postup do kvízové fáze bude muset vyplnit všechny loci položkami k zapamatování. Návrat z memorizační fáze bude možný kdykoliv, ale návrat ze scény reprezentující kvízovou fázi bude umožněn pouze po jejím vyhodnocení. Vyhodnocení bude také v kontrole uživatele a bude možné vyvolat bez nutnosti umístění všech položek k testování.

Nefunkční požadavky

- **NAM01: Prostředí bude velká místnost**

Prostředí bude reprezentováno velkou obytnou místností s akční plochou o rozloze alespoň 30 m^2 . Akční plochou se myslí prostor, do kterého je možné pokládat nábytek.

- **NAM02: V prostředí bude 10 loci**

Konečný počet loci (tvořen nábytkem) bude roven 10. Jeden nábytek může mít podle velikosti více loci.

- **NAM03: Jednotlivá loci budou ukládat položky po jednom**

Do loci nebude možné vkládat více položek k zapamatování/testování zároveň.

- **NAM04: Položky k zapamatování/testování budou reprezentovány 3D objekty**

Položky k zapamatování a testování budou hmatatelné objekty, které lze reprezentovat samostatným 3D objektem.

- **NAM05: Generování položek bude náhodné a z množiny o alespoň 15 objektech**

Položky k zapamatování a testování budou náhodně vybírány z množiny objektů, jejíž kardinalita je alespoň 15.

- **NAM06: Nábytek a položky k zapamatování/testování bude jednoduché rozšířit**

Architektura tříd objektů reprezentující nábytek a položky k zapamatování/testování bude umožňovat jednoduché přidávání dalších variant.

3.3 Výběr technologií

Ačkoliv je z požadavku NA01 zřejmé, že pro vývoj aplikace je zvolený UE4, slouží tato sekce k upřesnění detailů použití této technologie, které budou mít vliv na návrh tříd.

Vzhledem ke skutečnosti, že se pro autora této práce jedná o první projekt v Unreal Engine, bude veškerá logika aplikace tvořena za pomoci blueprintů. Konkrétním důvodem je rychlost prototypizace při vývoji, která by se měla přenést do rychlejšího poznání a pochopení herního frameworku¹.

Pro projekt nebude použita VR šablona, která je součástí UE4. Aplikace bude vytvořena z prázdného projektu v kombinaci se zásuvným modulem (tzv. plugin) VR Expansion. Tento

¹autorovi přijde systém herních objektů v UE4 méně intuitivní než např. ECS systém v Unity

plugin je open-source (MIT licence) a rozšiřuje engine o třídy, funkce a moduly, které podporují vývoj VR aplikací (např. podpora pro širokou škálu VR zařízení a přídavných aparátů, rozeznávání gest, podpora více hráčů atd.) [22].

Zamýšlený rozsah aplikace nevyžaduje většinu nástrojů tohoto pluginu – rozhraní, která se plánují použít v návrhu, by bylo možné nahradit snadno vlastními – přesto byl plugin zvolen pro případné rozšiřování aplikace. Příkladem je požadavek NA02, který specifikuje kompatibilitu se zařízením HTC Vive, ale díky pluginu je realizována alespoň částečná podpora i pro jiná zařízení², aniž by se na tato zařízení a jejich platformy cílilo při vývoji. Cílené rozšíření aplikace o další hardware či přidání podpory více hráčů³ by tedy nemělo způsobit příliš velké zásahy do architektury aplikace.

3.4 Návrh základních tříd

V interpretaci metody loci v kapitole 3.1 byly identifikovány položky k zapamatování/testování a nábytek jako nejpodstatnější entity pro návrh aplikace. Pro tyto entity je v této kapitole představen základní návrh tříd, který rozšiřuje základní třídy poskytované UE4 a pluginem VR Expansion. Zmíněná je i třída reprezentující entitu uživatele.

3.4.1 Třída *Item*

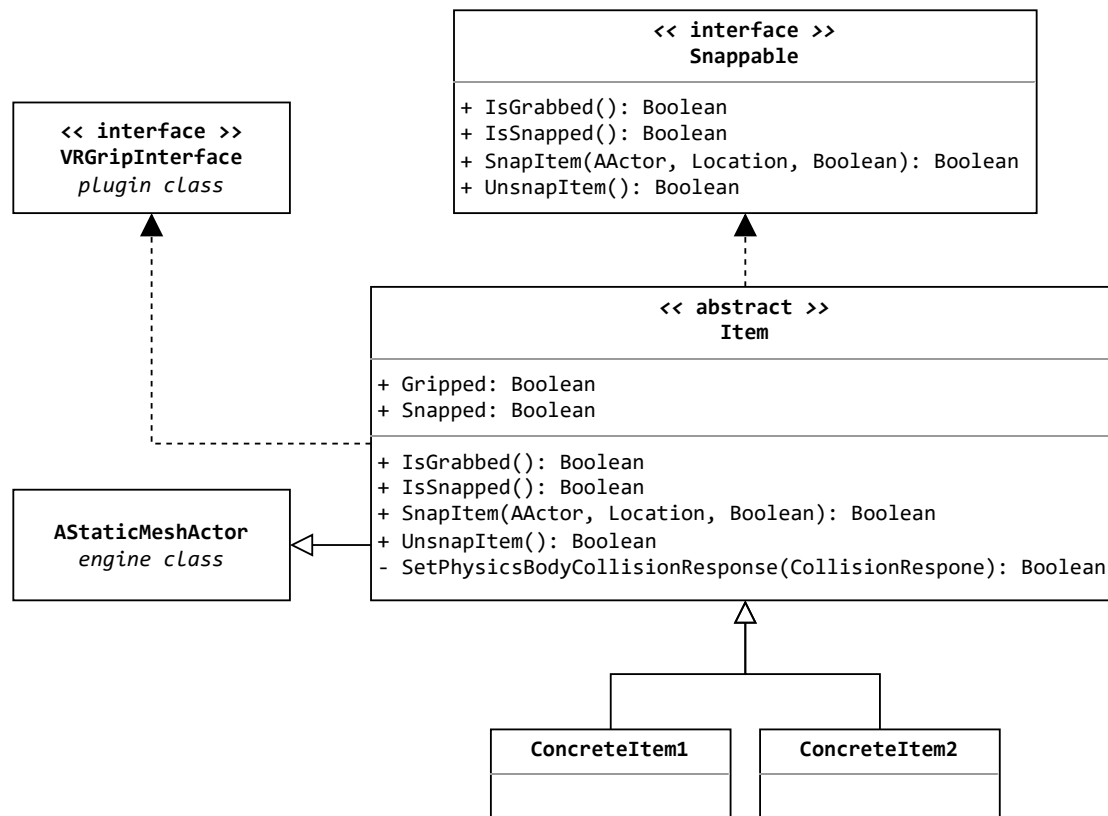
Třída *Item* reprezentuje položky k zapamatování a testování. Rozdíl mezi položkami k zapamatování a testování spočívá pouze v tom, ve které scéně se vygenerovaly – z hlediska třídního návrhu se pořád jedná o stejný objekt. Tato třída rozšiřuje třídu engineu *AStaticMeshActor*, jelikož se jedná o nejjednoduššího potomka třídy *AActor*, který zvládne samostatně reprezentovat 3D objekt. Třída také implementuje dvě rozhraní – *VRGripInterface* a *Snappable*. První zmiňované rozhraní je součástí pluginu a slouží ke generalizaci objektů, které uživatel aplikace bude moci přesouvat jejich uchopením. Rozhraní *Snappable* pak umožňuje objekty, které ho implementují, vkládat do loci.

Vzhledem k tomu, že *Item* je abstraktní třídou, je potřeba pro využití v aplikaci vytvořit její potomky (viz *ConcreteItem1* a *ConcreteItem2* na obrázku 3.1). Tyto potomky budou reprezentovat jednotlivé položky k zapamatování a testování, a to na úrovni tříd – neporovnává se adresa ukazatelů jednotlivých instancí, ale jejich příslušný datový typ. Porovnávání jednotlivých instancí je pragmatičtější řešení, které však není vhodné kvůli skutečnosti, že by si instance reprezentující stejný objekt nebyly rovné – problematické pro vyhodnocování ve kvízové fázi.

Všechny potomky třídy *Item* jsou v rámci aplikace v režii tříd *ItemManager* a *QuizItemManager*, které si udržují reference na veškeré běžící instance v kolekci. Tyto režijní třídy vychází ze stejného vstupního seznamu datových typů potomků, které reprezentují všechny položky k zapamatování/testování. Třída *ItemManager* v memorizační fázi náhodně vybere ze zmiňovaného seznamu deset datových typů, které následně začne generovat do prostředí, zatímco třída *QuizItemManager* v kvízové fázi postupně vygeneruje všechny položky ze seznamu.

²vyzkoušeno např. na Oculus Quest 2

³např. užití aplikace jako experimentální prostředí, ve kterém je uživatel školen výzkumným pracovníkem



■ **Obrázek 3.1** Diagram tříd popisující položky k zapamatování/testování

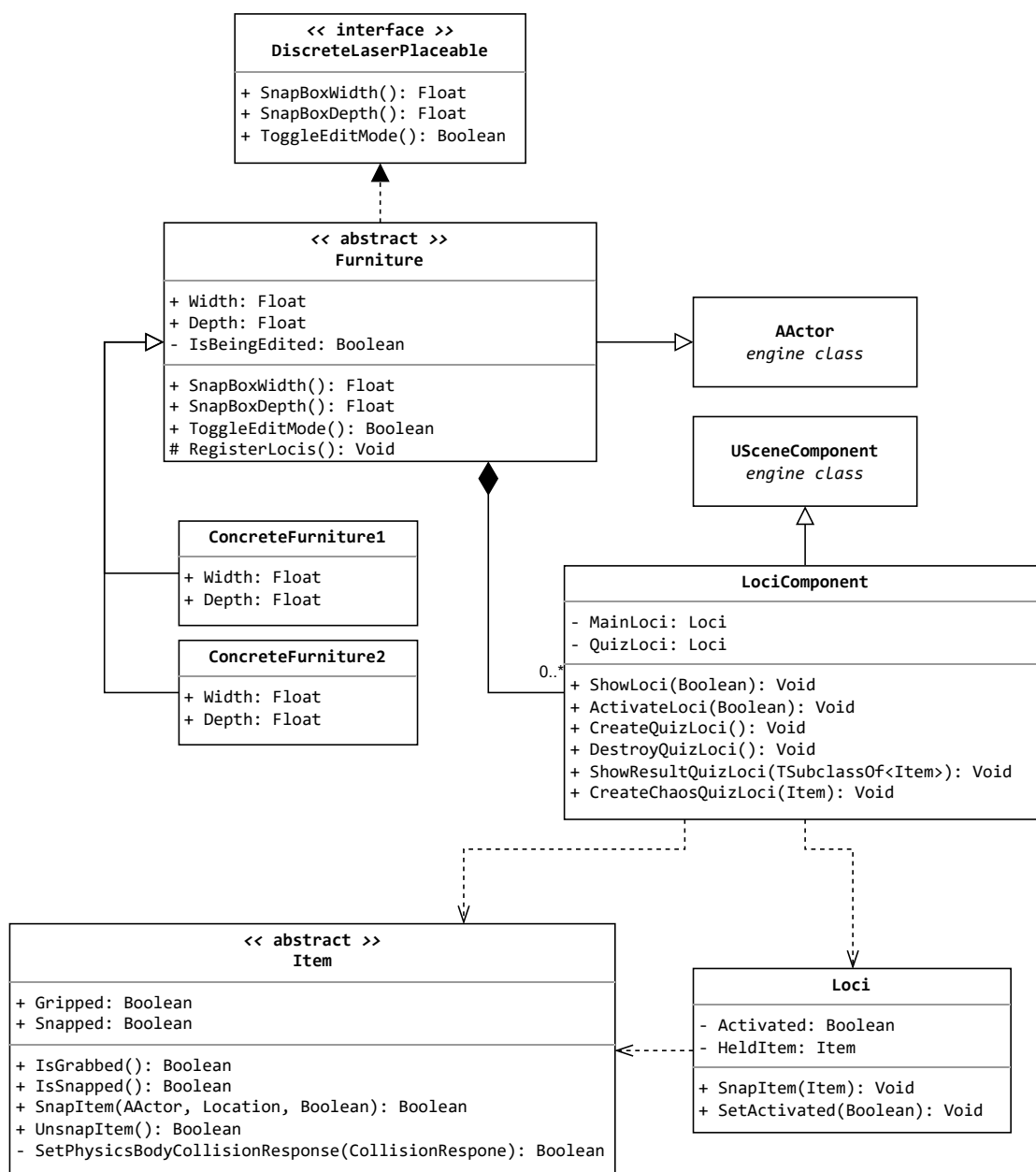
3.4.2 Třída Furniture

Třída *Furniture* dědí z třídy engineu *AActor* a představuje nábytek v prostředí. Součástí každého nábytku může být libovolný počet loci – toto je v návrhu realizováno agregací komponent *LociComponent*.

Tato komponenta dědí ze třídy engineu *USceneComponent* a na nábytku je reprezentována bodem, který slouží k zobrazení třídy *Loci* do prostředí. Součástí komponenty jsou dvě loci, jedno slouží k vložení potomka třídy *Item* během memorizační fáze, zatímco druhé se využívá pouze ve kvízové fázi. Komponenta porovnává v obou loci datové typy podtříd *Item* pro vyhodnocení kvízové fáze.

Rozhraní *DiscreteLaserPlaceable*, ze kterého třída *Furniture* dědí, umožňuje uživateli rozkládat nábytek v diskrétní mřížce za pomoci laserového indikátoru. Rozměry hloubky a šířky nábytku jsou určeny právě pro výpočty transformací ve zmiňované mřížce.

Furniture je také jako *Item* abstraktní třídou a je tedy nutné vytvořit potomky pro každý unikátní kus nábytku. Všechny takové podtřídy jsou předány režijní třídě *FurnitureManager*, která se stará nejen o jejich náhodné generování do prostředí během familiarizační fáze, ale i o zobrazení všech loci v memorizační fázi, vytvoření kvízových loci a jejich případné vyhodnocení během kvízové fáze.



■ Obrázek 3.2 Diagram tříd popisující nábytek a loci

3.4.3 Třída *PlayerCharacter*

Třída *PlayerCharacter* je implementačně specifická a není možné vytvořit její kompletní návrh. Předpokládá se, že bude podtřídou třídy pluginu *VRCharacter* a že bude obsahovat tři komponenty: *MovementComponent*, *GrippingComponent* a *DiscreteLaserPlacingComponent*. První komponenta bude implementovat různé režimy pohybu, zatímco poslední dvě komponenty slouží k manipulaci potomků tříd *Item* a *Furniture*.

Implementace řešení

Vzhledem k tomu, že při tvorbě projektu nebyla v enginu zvolena VR šablona, je potřeba upravit nastavení projektu tak, aby se spouštěla v headsetu. V nastavení projektu je tedy nutné zaškrtnout políčko „Start in VR“. Dále je nutné nainstalovat plugin, buďto předkompilovanými balíčky, nebo manuálním zkompilováním do enginu. Správně nainstalovaný plugin se bude zobrazovat mezi ostatními zásuvnými moduly.

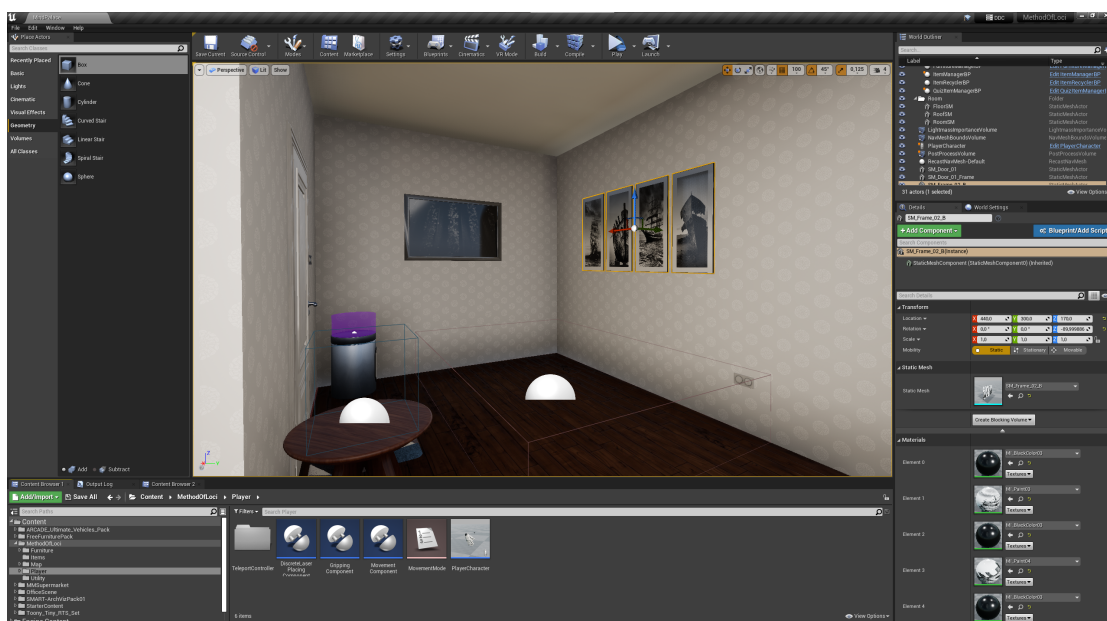
Během implementace řešení jsou pro grafickou reprezentaci aplikace použity asset balíčky z Epic Store, které se na dané platformě každý měsíc objevují zdarma. Jediné použité assety, jejichž součástí je nějaká logika, jsou blueprinty tlačítek a ovladačů ze šablony pluginu. Tyto blueprinty jsou rozšířením blueprintů z VR šablony enginu.

4.1 Prostředí

Největší část prvního dojmu bude tvořit podoba místnosti, ve které se budou odehrávat všechny scény. To se v UE4 vytváří tzv. levelem, tedy úrovní aplikace, kterou je nutné přidat v prohlížeči obsahu (okno s názvem Content Browser) – k tomu slouží zelené tlačítko „Add/Import“. Po přidání zmiňované úrovně je do ní nutné přidat základní osvětlení, a to kombinací nějakých enginem předem připravených objektů (např. *SkyLight* a *DirectionalLight*).

4.1.1 Místnost

Po přidání úrovně je potřeba vytvořit statickou mesh, která bude reprezentovat místnost. K tomu poslouží nástroj Box (v editoru Level Editor), kterým se do úrovně přidá editovatelná geometrie. Tuto geometrii je možné tvarovat v panelu detailů na pravé straně editoru – dokonce je možné kombinovat více takových geometrií množinovými operacemi (např. pro podlahu místnosti byly množinovým sjednocením spojené dva tvary o rozměrech $600 \times 600 \times 10$ a $300 \times 300 \times 10$, kde jednotka enginu představuje 1 cm v reálném světě). K uložení do statické meshe stačí označit všechny cílové geometrie a stisknout v panelu detailů na tlačítko „Create Static Mesh“. Taková statická mesh se uloží bez kolizí, ty je nutné přidat v editoru Static Mesh Editor.



■ **Obrázek 4.1** Ukázka z tvorby úrovně reprezentující metodu loci

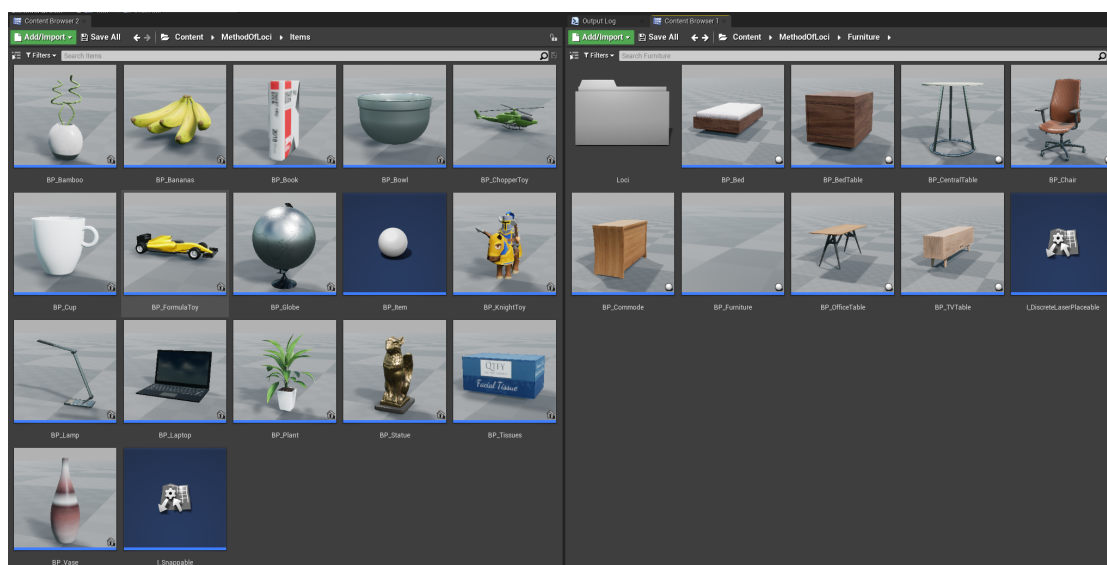
Nově vytvořenou místnost je nyní možné graficky přizpůsobit přidáním materiálům k jednotlivým statickým meshům, přidáním dekorativních objektů či vytvořením vnitřního osvětlení (viz obrázek 4.1). Do místnosti je také nutné přidat třídu engineu *NavMeshBoundsVolume*, která umožní uživateli pohyb s automatickým naváděním (např. teleportace).

4.1.2 Objekty scén

Výběr objektů reprezentující položky k zapamatování/testování a nábytku spočíval v prohledávání bezplatných balíčků assetů, které autor práce nasbíral za poslední rok. Asset byl vybrán, pokud se jevil jako tématicky vhodný a pokud zároveň představoval v kontextu ostatních objektů dostatečnou rozmanitost (viz obrázek 4.2, na levé straně se nachází položky k zapamatování/-testování a na pravé nábytek).

Pro nábytek bylo vybráno sedm objektů tak, aby dohromady utvořily deset loci. Seznam zvolených objektů tvoří: postel, noční stolek, konferenční stolek, kancelářská židle, komoda, psací stůl a televizní stolek. Ve skutečnosti jsou všechny loci tvořeny pouze pěti kusy nábytku, zatímco postel a židle slouží jako rekvizity.

Ačkoliv se do scény bude generovat pouze deset položek k zapamatování (stejný počet jako loci), je jedním z požadavků generování z kolekce o alespoň patnácti položkách. Složení této kolekce je následující: bambusová rostlina, banány, kniha, mísa, model vrtulníku, hrnek, model formule, glóbus, rytíř na koni, stolní lampa, laptop, listnatá rostlina, socha, kapesníky a váza.



■ **Obrázek 4.2** 3D objekty reprezentující položky k zapamatování/testování a nábytek

4.1.3 Logika

Implementace logiky aplikace stavěla na navrženém základu z kapitoly 3.4, který byl dodržen a případně jen rozšířen – např. skripty, které mění barvu loci podle obsazení či vyhodnocení po kvízové části. Nejvíce klíčové pro logiku aplikace z hlediska implementace jsou blueprints reprezentující třídy *ApplicationManager* a *PlayerCharacter*.

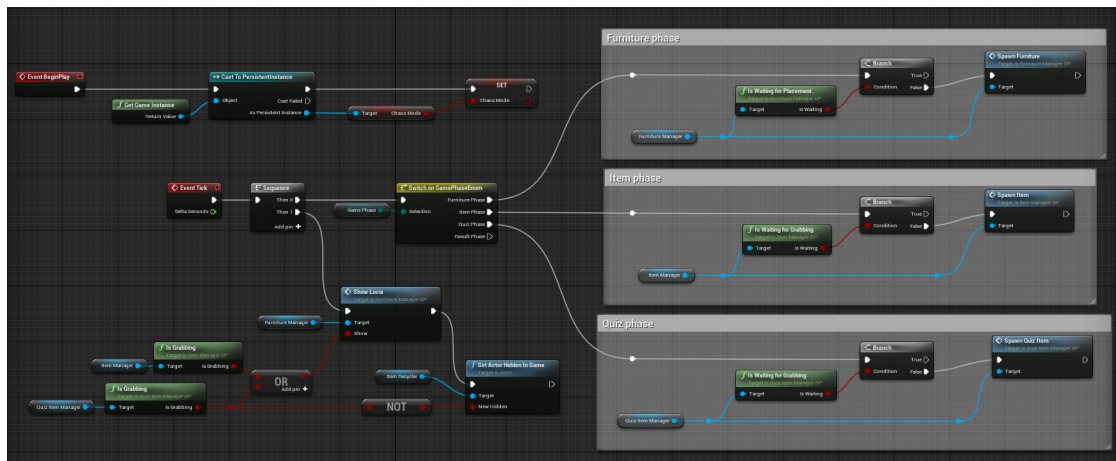
Třída *ApplicationManager*

Jedná se o stěžejní třídu řídicí logiku scén v úrovni, jejíž součástí jsou všechny ostatní režijní třídy jako *FurnitureManager* či *ItemManager*. Chování třídy *ApplicationManager* je podobné konečnému automatu – podle aktuální scény deleguje práci na zodpovědnou režijní třídu (např. v memorizační fázi požaduje generování položek po *ItemManager*) a kontroluje splnění podmínek pro přechod do další či předchozí scény.

Součástí třídy *ApplicationManager* je třída *ItemRecycler*, která v kvízové fázi metody loci slouží jako odpadkový koš pro položky, u kterých se uživatel rozhodne, že nepatří mezi memorizované položky. Tento koš si udržuje vlastní kolekci recyklovaných položek, která se předá třídě *QuizItemManager* v případě, že už je prázdná, aby scéna mohla generovat položky dál.

Třída *PlayerCharacter*

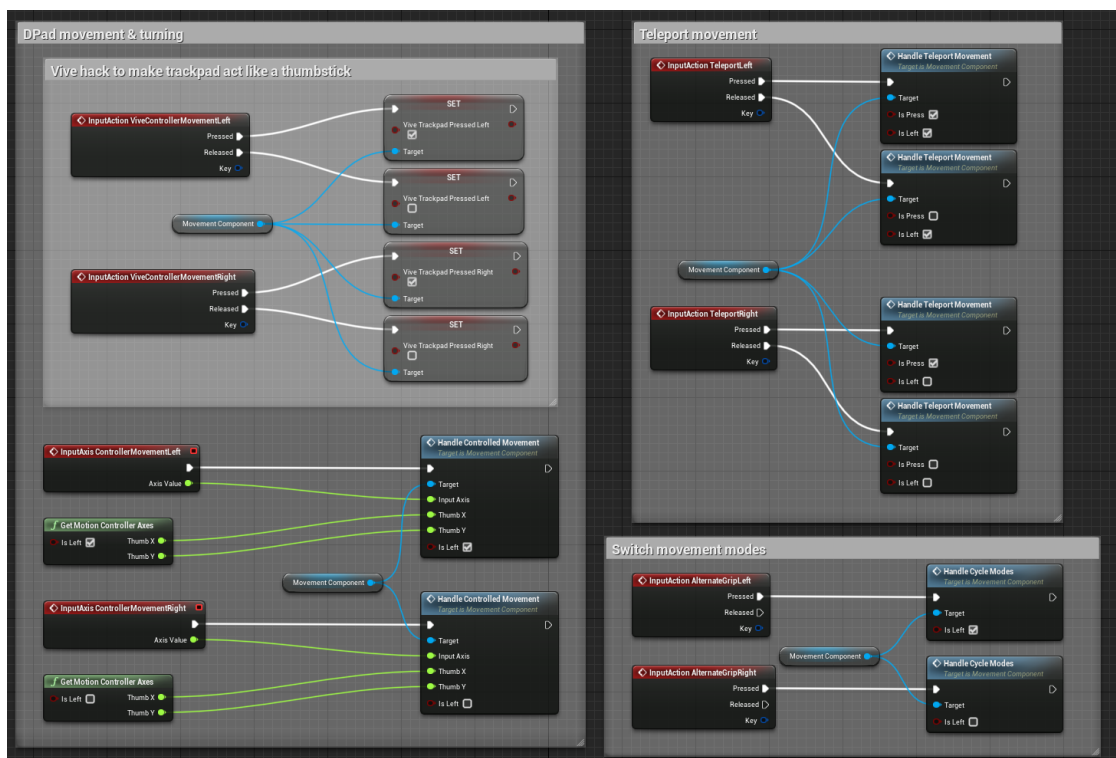
Třída implementuje komponenty zmiňované v kapitole 3.4. Komponenta *MovementComponent* implementuje čtyři různé režimy pohybu. Režimy „Teleport“ a „Navigate“ využívají *NavMeshBoundsVolume* pro automatickou navigaci, zatímco režimy „Oriented (Dpad)“ (využívající směrových vektorů ovladačů) a „Oriented (HMD)“ (využívající směrového vektoru headsetu) jsou manuální. Tato komponenta rozlišuje dominantní ruku od nedominantní (nedominantní ruka má



■ Obrázek 4.3 Logika delegace režie v bluepintu ApplicationManager

dva režimy pro rotaci uživatele). Režimy rotace se liší pouze v plynulosti (pro případ kdyby se uživateli dělalo špatně z replikovaných pohybů).

Další komponenta, která dělí funkce pro obě ruce, je *DiscreteLaserPlacingComponent*. Dominantní ruka slouží k přepínání editace polohy nábytku, na který je mířeno laserem, zatímco nedominantní ruka tento nábytek otáčí o 90 stupňů. Poslední komponenta *GrippingComponent* mezi rukami nedělá rozdíl a umožňuje tedy uchopení položek libovolným ovladačem.

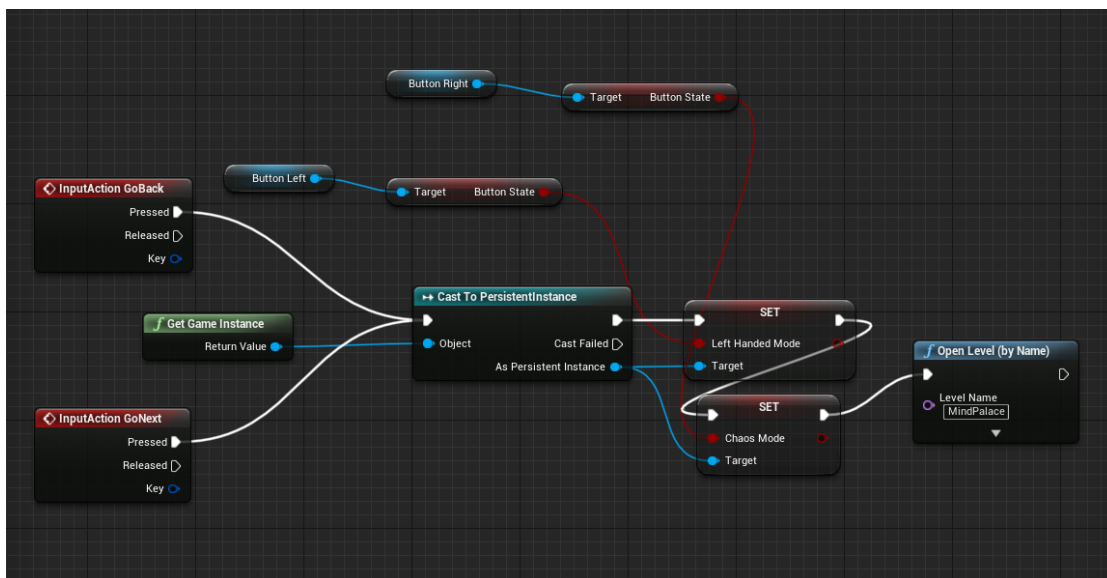


■ Obrázek 4.4 Použití komponent v bluepintu PlayerCharacter

4.2 Intro

Pro aktivování levorukého režimu nebo chaotické varianty kvízové fáze metody loci je potřeba získat vstup od uživatele (tj. zda si tyto volby přeje aktivovat, a to nejlépe po spuštění samotné aplikace). Toto je realizováno přidáním úvodní scény, která obsahuje tlačítka pro aktivování jednotlivých voleb. Stejně jako v kapitole 4.1 je vytvořena a přidána nová úroveň se základními náležitostmi, jako je osvětlení a podlaha.

Pro tlačítka je použit blueprint *ButtonActor* ze šablony pluginu. Tlačítko udržuje stav zmáčknutí a do úvodní úrovně aplikace tedy postačí vložit dvě instance. Obě tlačítka pak budou referována v blueprintu *IntroCharacter*, který má téměř stejnou strukturu jako blueprint postavy hráče v úrovni reprezentující scény metody loci, pouze neobsahuje žádnou logiku (kromě přepnutí do úrovně reprezentující metodu loci).



■ **Obrázek 4.5** Logika přepnutí úrovně v blueprintu *IntroCharacter*

Přepnutí úrovně probíhá funkcí *OpenLevel*, do které stačí předat jméno cílového levelu. Je důležité zmínit, že takto se volby režimů do další úrovně nepřenesou, to je nutné zařídit přes tzv. herní instanci, jedná se o instanci (viz obrázek 4.5), která je perzistentní napříč celou aplikací. V této instanci je tedy možné vytvářet proměnné, jejichž nastavenou hodnotu lze pak načíst v počáteční události nové úrovně.

Test řešení

V této kapitole proběhne porovnání efektivity virtuální implementace metody loci s její tradiční podobou, a to z hlediska autora práce. V rámci porovnání proběhnou dva pokusy – memorizace množiny objektů a memorizace posloupnosti objektů, které se zabývají efektivitou zapamatování deseti objektů. Oba pokusy popisují selekci prostředí, jaký soubor objektů byl pro zapamatování vygenerován a jak si autor vedl při vybavování těchto objektů v testech. Testy memorizace probíhají 1 hodinu a 24 hodin po samotném zapamatování.

Ve virtuální implementaci metody loci jsou všechny kroky uskutečněny uvnitř aplikace. Pro tradiční variantu je náhodný výběr objektů pro memorizaci uskutečněn webovým nástrojem¹, který náhodně vybere deset položek ze vstupního seznamu. Tento vstupní seznam je složen ze stejných objektů, které se generují v aplikaci (viz kapitola 4.1.2). Pro testy je použit stejný nástroj, ale je generovaná náhodná permutace se všemi objekty.

Implementovaná interpretace metody loci umožňuje další způsob používání aplikace pro pokusy, a to díky chaotickému režimu – v testovací fázi aplikace se rozmístí náhodné objekty do jednotlivých loci. Tento režim je ovšem špatně replikovatelný v tradiční podobě metody, a proto je pro oba pokusy zvolen režim klasický.

5.1 Memorizace množiny objektů

5.1.1 Virtuální metoda loci

Prostředí a loci

V naimplementované aplikaci je prostředí tvořeno ve familiarizační fázi – fáze, v níž uživatelé pokládají nábytek. Pro memorizaci množiny objektů si autor rozložil nábytek způsobem znázorněným na obrázku 5.1.

¹randomlists.com



■ **Obrázek 5.1** Rozložení nábytku ve virtuální místnosti

Objekty a jejich memorizace

Aplikace vygenerovala v memorizační fázi následujících deset objektů: rytíř na koni, bambusová rostlina, stolní lampa, hrnek, kapesníky, socha, listnatá rostlina, váza, model formule a banány.

Vzhledem k tomu, že při tomto testu nezáleží na pořadí objektů, mohl autor při pokládání objektů do dílčích loci jednat dle vlastního uvážení. Autor tedy sdružoval objekty s loci na základě vlastních tendencí, např. stolní lampa byla položena na levou část stolu, protože autor píše pravou rukou. I s četným prohazováním objektů mezi loci byl autor s memorizací hotov během 3 minut a 34 vteřin.

Test memorizace

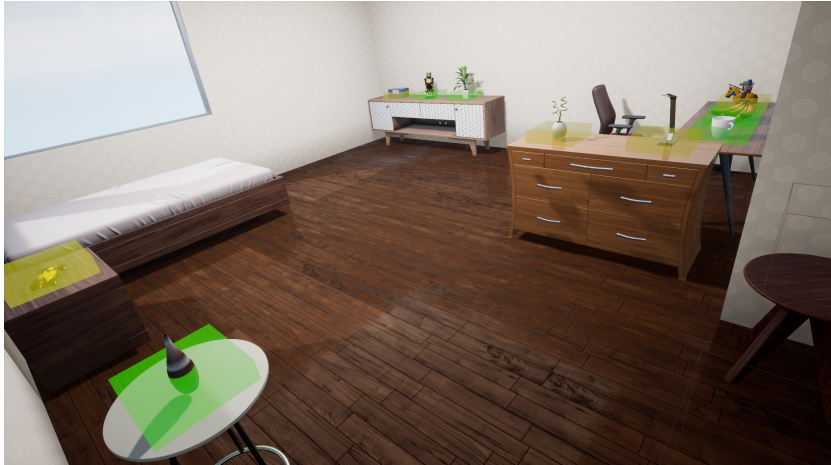
Autor si vedl bezchybně při prvním testu memorizace, a to v čase 1 minuta a 12 vteřin. Pro tento pokus je kvízová fáze aplikace velice rychlá, protože není nutné umisťovat objekty do jejich původní polohy, stačí je pouze vložit do libovolného loci za předpokladu, že objekt je součástí memorizované množiny (viz obrázek 5.2, žlutě označené objekty patří do loci, ale jsou na odlišném místě, zeleně jsou pak označeny objekty, které jsou i na správné pozici).

Druhý test v aplikaci, po 24 hodinách od zapamatování množiny objektů, proběhl téměř identicky jako ten první. Autor bez potíží a chyb dokončil kvízovou fázi za 1 minutu a 16 vteřin.

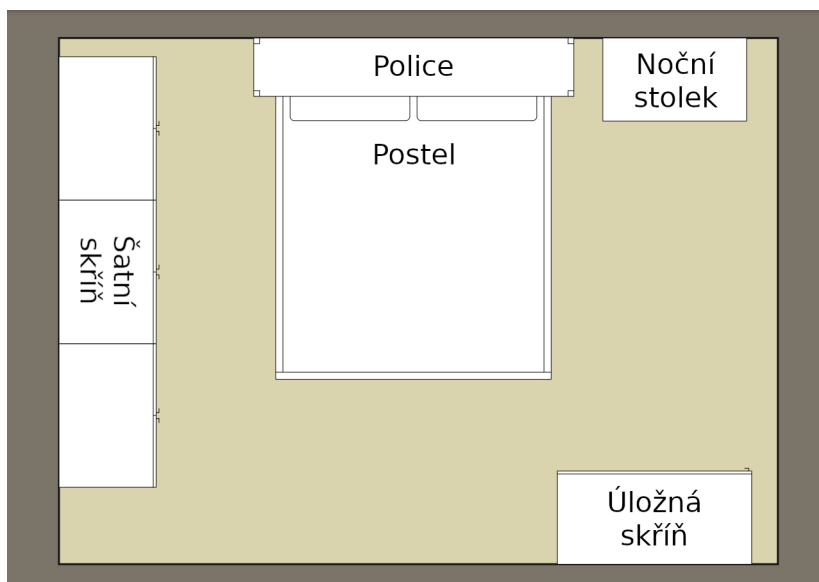
5.1.2 Tradiční metoda loci

Prostředí a loci

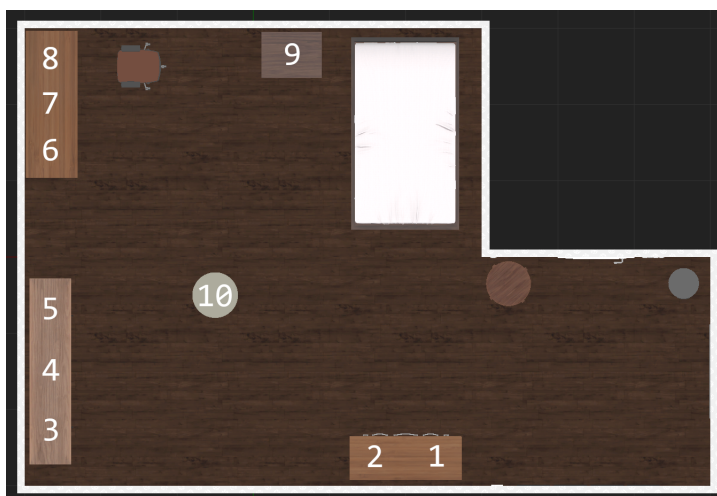
Pro tradiční variantu memorizace množiny objektů si autor zvolil svoji ložnici, ve které použil nábytek jako loci (viz obrázek 5.3). Místnost má přibližnou rozlohu 20 m^2 a neobsahuje stejný počet nábytku jako loci, proto autor rozdělil šatní skříň, úložnou skříň a postel do více částí.



■ Obrázek 5.2 Vyhodnocení kvízové fáze aplikace



■ Obrázek 5.3 Náčrt ložnice s nábytkem



■ **Obrázek 5.4** Rozložení nábytku a loci ve virtuální místnosti (včetně pořadí)

Objekty a jejich memorizace

Webovým nástrojem byly k memorizaci vygenerovány následující položky: socha, model formule, stolní lampa, model vrtulníku, hrnek, kniha, kapesníky, rytíř na koni, váza a listnatá rostlina.

Objekty byly do mentální podoby ložnice vkládány opět dle autorovy tendence, např. na polici postavil stolní lampu a do postele si odložil kapesníky s knihou. Memorizace objektů netrvala déle než 8 minut a 19 vteřin.

Test memorizace

První test memorizace zvládl autor perfektně – vygenerovaný list objektů správně rozdělil na objekty patřící a nepatřící do množiny zapamatovaných objektů. Autor nezaváhal u jediné položky a test dokončil za 28 vteřin.

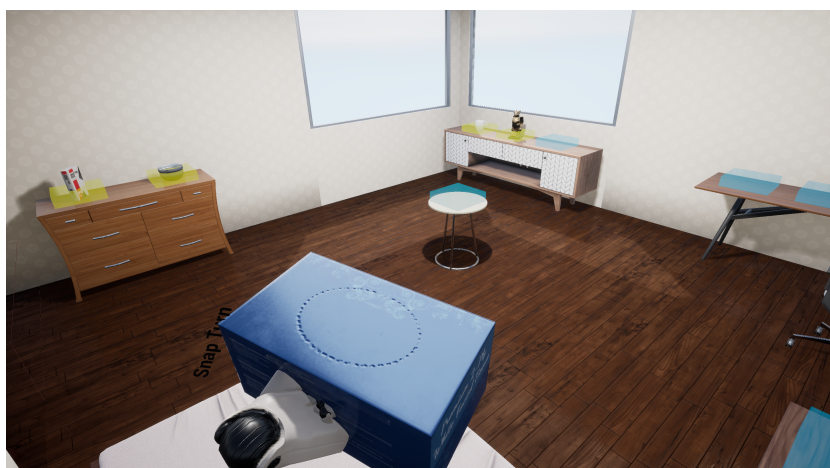
Druhý test se už neobešel bez menšího zaváhání – u vygenerovaného listu autor znejistil u bambusové a listnaté rostliny. Nakonec zvolil správnou položku (listnatá rostlina) a test dokončil bez chyb v 50 vteřinách.

5.2 Memorizace posloupnosti objektů

5.2.1 Virtuální metoda loci

Prostředí a loci

Na obrázku 5.4 je z ptáčí perspektivy znázorněna místnost s nábytkem, který si autor rozložil během familiarizační fáze interaktivní implementace. Autor stanovil, že se prostředím prochází ve směru hodinových ručiček a začíná se u vstupu. Pro referenci slouží očíslování jednotlivých loci na zmiňovaném obrázku.



■ **Obrázek 5.5** Postupné ukládání objektů do loci ve virtuální místnosti

Objekty a jejich memorizace

Memorizační fáze interaktivní implementace generovala postupně následující objekty: kniha, mísa, hrnek, socha, kapesníky, banány, stolní lampa, model formule, bambusová rostlina a váza.

Vygenerované objekty byly poskládány autorem do jednotlivých loci v pořadí jejich objevení na scéně (viz obrázek 5.5). Autor memorizací strávil 4 minuty a 27 vteřin.

Test memorizace

V kvízové fázi interaktivní implementace bylo (po hodině od memorizační fáze) ověřeno úspěšné zapamatování všech předmětů, a to v čase 1 minuta a 10 vteřin. Z postupně vygenerovaných objektů byly autorem vybrány ty správné, které byly následně umístěny do svých příslušných loci. Autor zaváhal pouze u hrnku, který nejdříve položil do loci č. 9 namísto loci č. 3. Tato chyba byla opravena ve chvíli, kdy se na scéně objevila bambusová rostlina, která patří právě do loci č. 9.

Autor následujícího dne dosáhl bezchybného výsledku v čase 1 minuta a 7 vteřin. Výsledek je však zkreslený skutečností, že si autor připomněl posloupnost zapamatovaných objektů při tvorbě obrázku 5.4 pro tuto práci.

5.2.2 Tradiční metoda loci

Prostředí a loci

Pro tradiční podobu metody loci autor zvolil cestu reprezentující vynášení odpadků. Cesta je rozdělena na dílčí kroky, které jsou autorem vždy prováděny při vykonávání zmiňované domácí práce. Loci tvoří fyzické lokace, ve kterých dané kroky probíhají:

1. *získání klíčů od domu* - pracovní stůl v obývacím pokoji
2. *vyjmutí odpadků z odpadkového koše* - malá předsň

3. *obutí bot uklizených v botníku* - velká předsíň
4. *zavření dveří do domu* - vchodové poschodí
5. *vložení odpadků do smíšeného odpadu* - zahrada za domem
6. *otevření dveří do domu* - vchodové poschodí
7. *uklizení bot do botníku* - velká předsíň
8. *vložení pytle do odpadkového koše* - malá předsíň
9. *umytí rukou* - koupelna
10. *odložení klíčů od domu* - pracovní stůl v obývacím pokoji

Objekty a jejich memorizace

Položky k zapamatování byly vygenerovány v následujícím pořadí: stolní lampa, bambusová rostlina, hrnek, listnatá rostlina, model vrtulníku, váza, glóbus, banány, socha a rytíř na koni.

Pro zapamatování objektů v posloupnosti začal autor procházet mentální podobu prostředí po různých částech, aby se ujistil, že nevynechává některá loci. Poté začal procházet prostředím souvisle, dokud jím neprošel bez chyb a váhání – celkově dva průchody. Zapamatování trvalo 8 minut a 13 vteřin.

Test memorizace

Po hodině od zapamatování všech položek v pořadí si autor vzal prázdný papír a vygeneroval testovací posloupnost objektů. Postupně k jednotlivým objektům z paměti vyplňoval jejich pořadí v prostředí. Jestliže objekt do zapamatované posloupnosti nepatřil, přeškrtl jej. Autor všechny položky správně očísloval během 52 vteřin.

Autor tento postup replikoval následující den, kde s časem 1 minuta a 27 vteřin úspěšně sepsal všechny objekty v daném pořadí. Je však nutné zmínit, že autor zaváhal s pořadovým číslem glóbusu (velká předsíň) a vědomě započal imaginární cestu po prostředí. Ačkoliv se nutně nejedná o chybu, poukazuje to na tendenci autora používat úplný mentální průchod prostředím pouze v momentě, kdy si není jistý svou odpovědí.

5.3 Výsledky

Z dat v tabulce 5.1 je možné vyvodit, že memorizování objektů je snadnější či alespoň rychlejší ve virtuální implementaci metody loci, jelikož v nich autor strávil memorizací mnohem méně času (a to v obou pokusech). Autor však tuto skutečnost považuje spíše za „herní efekt“ prostředí na krátkodobou paměť a dále poznamenává, že virtuální prostředí z dlouhodobé paměti rychle zmizí, není-li pravidelně připomínáno. Bez prostředí a loci pak chybí hlavní asociace k zapamatovaným položkám.

Porovnání dat z plnění testů memorizace neprokázalo zásadní rozdíl mezi jednotlivými variantami a lze tedy konstatovat, že jejich efektivita je z hlediska autora téměř identická. Je však

■ **Tabulka 5.1** Shrnutí dob trvání obou pokusů (v minutách)

	Virtuální metoda loci	Tradiční metoda loci
Memorizace množiny objektů	3:34	8:19
Test po 1 hodině	1:12	0:28
Test po 24 hodinách	1:16	0:50
Memorizace posloupnosti objektů	4:27	8:13
Test po 1 hodině	1:10	0:52
Test po 24 hodinách	1:07	1:27

■ **Tabulka 5.2** Porovnání vlastností variant metody loci

Vlastnosti	Virtuální metoda loci	Tradiční metoda loci
Použití známého prostředí	Ano	Ano
Využití prostorových asociací	Ano	Ano
Lze použít k zapamatování množiny objektů	Ano	Ano
Lze použít k zapamatování posloupnosti objektů	Ano	Ano
Lze použít bez technického vybavení	Ne	Ano
Lze použít bez znalosti metody loci	Ano	Ne
Lze použít k zapamatování abstraktních konceptů	Ne	Ano
Umožňuje vlastní způsob zakódování informací do loci	Ne	Ano
Umožňuje okamžitý test zapamatování	Ano	Ne
Flexibilní počet položek k zapamatování	Ne	Ano

nutné zdůraznit, že se varianty liší ve svých možnostech (viz tabulka 5.2) – flexibilitu metody loci není prozatím možné plnohodnotně převést do virtuální podoby.

Závěr

Metoda loci je velice starou mnemotechnickou pomůckou, ale její efektivita na paměť je nepopíratelná, a to i přestože žádné dosavadní studie nebyly schopny vysvětlit, v čem síla této metody spočívá. Výzkumníci začali k dalšímu zkoumání metody vytvářet její interaktivní varianty, prozatím stále bez větších pokroků a stěžejních výsledků.

Hlavním cílem této práce bylo vytvořit (na základě analýzy zmiňované problematiky) vlastní návrh a implementaci interaktivní metody loci takovým způsobem, aby výsledná aplikace případně mohla tuto mnemotechnickou pomůcku představit širšímu publiku. Součástí aplikace jsou všechny kroky, které jsou obvyklé spjaté s memorizací užití této metody, včetně možnosti ověřit si správné zapamatování v podobě kvízové minihry. Kvízová minihra má navíc alternativu v podobě „chaotického režimu“. Při vývoji byl kladen důraz na interaktivní vlastnosti zařízení pro virtuální realitu a na případnou možnost aplikaci rozšířit. Aplikace podporuje jak pravoruké, tak i levoruké ovládání.

Implementace aplikace probíhala v systému Unreal Engine 4, který byl v práci představen a jehož užití bylo popsáno jak v návrhové fázi, tak i v samotné implementaci. Přestože byl během vypracovávání této práce publikován Unreal Engine 5, jsou návrh i implementace stále relevantní, a to díky zvoleným postupům při vývoji, které by neměl být problém replikovat v nové generaci enginu.

Na konci práce proběhlo porovnání výsledné aplikace s tradiční podobou metody loci. Porovnávala se efektivita z hlediska autora práce, a to ve dvou nezávislých pokusech. Jeden z pokusů se zabýval memorizací množiny položek, zatímco druhý pokus se zabýval memorizací posloupnosti položek. V obou pokusech nebyl zaznamenán zásadní rozdíl v efektivitě jednotlivých variant metody loci.

V budoucnosti by bylo možné rozšířit aplikaci o další způsoby testování memorizace či o funkcionality, které by umožnily užití aplikace pro výzkumné účely a umožnit tak například výzkum zabývající se porovnáváním virtuální metody loci s tradiční variantou u jednotlivců s nízkou představivostí, či dokonce afantazií.

Bibliografie

1. LEGGE, Eric L.G. et al. Building a memory palace in minutes: Equivalent memory performance using virtual versus conventional environments with the Method of Loci. In: *Acta Psychologica* [online]. 2012, sv. 141, s. 380–390 [cit. 2022-04-19]. ISSN 0001-6918. Dostupné z DOI: <https://doi.org/10.1016/j.actpsy.2012.09.002>.
2. HEEREMA, Esther. *What Is the Method of Loci?* [online]. 2022 [cit. 2022-04-19]. Dostupné z: <https://www.verywellhealth.com/will-the-method-of-loci-mnemonic-improve-your-memory-98411>.
3. DALGLEISH, Tim et al. Method-of-Loci as a Mnemonic Device to Facilitate Access to Self-Affirming Personal Memories for Individuals With Depression. In: *Clinical Psychological Science* [online]. 2013, sv. 1, s. 156–162 [cit. 2022-04-19]. ISSN 2167-7034. Dostupné z DOI: [10.1177/2167702612468111](https://doi.org/10.1177/2167702612468111).
4. MAGUIRE, Eleanor et al. Routes to Remembering: The Brains Behind Superior Memory. In: *Nature neuroscience* [online]. 2003, sv. 6, s. 90–95 [cit. 2022-04-22]. ISSN 1546-1726. Dostupné z DOI: [10.1038/nn988](https://doi.org/10.1038/nn988).
5. *Learning Foreign Languages* [online]. ©1999-2022 [cit. 2022-04-22]. Dostupné z: <https://www.academictips.org/memory/forlangu.html#TownLanguage>.
6. PEETERS, Anco; SEGUNDO-ORTIN, Miguel. Misplacing memories? An enactive approach to the virtual memory palace. In: *Consciousness and Cognition* [online]. 2019, sv. 76, s. 1–12 [cit. 2022-04-19]. ISSN 1053-8100. Dostupné z DOI: <https://doi.org/10.1016/j.concog.2019.102834>.
7. KROKOS, Eric; PLAISANT, Catherine; VARSHNEY, Amitabh. Virtual memory palaces: immersion aids recall. In: *Virtual Reality* [online]. 2019, sv. 23, s. 1–15 [cit. 2022-04-24]. ISSN 1434-9957. Dostupné z DOI: [10.1007/s10055-018-0346-3](https://doi.org/10.1007/s10055-018-0346-3).
8. DENHAM, Thomas. *What is Unreal Engine?* [online]. ©2022 [cit. 2022-04-27]. Dostupné z: <https://conceptartempire.com/what-is-unreal-engine/>.
9. NAHODIL, Martin. *CD Projekt vysvětluje, proč pro vývoj nového Zaklánače zvolil Unreal Engine 5* [online]. 2022 [cit. 2022-04-27]. Dostupné z: <https://doupe.zive.cz/clanek/cd-projekt-vysvetluje-proc-pro-vyvoj-noveho-zaklinace-zvolil-unreal-engine-5>.

10. *Unreal Engine commercial game deployment guidelines* [online]. Epic Games, ©2004-2022 [cit. 2022-04-27]. Dostupné z: <https://www.unrealengine.com/en-US/release>.
11. *Unreal Engine* [online]. 2022 [cit. 2022-04-27]. Dostupné z: https://en.wikipedia.org/wiki/Unreal_Engine.
12. *Contributing to Unreal Engine 4* [online dokumentace]. Epic Games, ©2004-2022 [cit. 2022-04-27]. Dostupné z: <https://docs.unrealengine.com/4.27/en-US/ProgrammingAndScripting/ProgrammingWithCPP/DownloadingSourceCode/ContributingToUnreal/>.
13. *Virtual Reality Development* [online dokumentace]. Epic Games, ©2004-2022 [cit. 2022-04-27]. Dostupné z: <https://docs.unrealengine.com/4.27/en-US/SharingAndReleasing/XRDevelopment/VR/>.
14. *Steam VR Prerequisites* [online dokumentace]. Epic Games, ©2004-2022 [cit. 2022-04-27]. Dostupné z: <https://docs.unrealengine.com/4.27/en-US/SharingAndReleasing/XRDevelopment/VR/VRPlatforms/SteamVR/Prerequisites/>.
15. FORSYTHE, Alex. *Blueprints vs. C++: How They Fit Together and Why You Should Use Both* [online]. YouTube [video], 2021 [cit. 2022-04-27]. Dostupné z: <https://www.youtube.com/watch?v=VMZftEVDuCE>.
16. *Balancing Blueprint and C++* [online dokumentace]. Epic Games, ©2004-2022 [cit. 2022-05-01]. Dostupné z: <https://docs.unrealengine.com/4.27/en-US/Resources/SampleGames/ARPG/BalancingBlueprintAndCPP/>.
17. *UnrealHeaderTool* [online dokumentace]. Epic Games, ©2004-2022 [cit. 2022-05-01]. Dostupné z: <https://docs.unrealengine.com/4.27/en-US/ProductionPipelines/BuildTools/UnrealHeaderTool/>.
18. *Unreal Object Handling* [online dokumentace]. Epic Games, ©2004-2022 [cit. 2022-05-01]. Dostupné z: <https://docs.unrealengine.com/4.27/en-US/ProgrammingAndScripting/ProgrammingWithCPP/UnrealArchitecture/Objects/Optimizations/>.
19. *Introduction to Blueprints* [online dokumentace]. Epic Games, ©2004-2022 [cit. 2022-05-01]. Dostupné z: <https://docs.unrealengine.com/4.27/en-US/ProgrammingAndScripting/Blueprints/GettingStarted/>.
20. *Gameplay Framework Quick Reference* [online dokumentace]. Epic Games, ©2004-2022 [cit. 2022-05-01]. Dostupné z: <https://docs.unrealengine.com/4.27/en-US/InteractiveExperiences/Framework/QuickReference/>.
21. *Tools and Editors* [online dokumentace]. Epic Games, ©2004-2022 [cit. 2022-04-28]. Dostupné z: <https://docs.unrealengine.com/4.27/en-US/Basics/ToolsAndEditors/>.
22. STATZER, Joshua. *About* [online]. [B.r.] [cit. 2022-05-05]. Dostupné z: <https://vreue4.com/>.

Obsah přiloženého média

readme.txt	stručný popis obsahu média
exe	adresář se spustitelnou formou implementace
src		
_ MethodOfLoci	zdrojový projekt implementace
_ thesis	zdrojová forma práce ve formátu \LaTeX
text	text práce
_ thesis.pdf	text práce ve formátu PDF
vid	adresář s ukázkou běhu programu