

Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Josef Reidinger

Episodická paměť pro postavy z IVE

Kabinet software a výuky informatiky

Vedoucí bakalářské práce: Mgr. Cyril Brom

Studijní program: Informatika, Programování

2007

Rád bych na tomto místě poděkoval Mgr. Cyrilu Bromovi za vedení a usměřování práce. Další poděkování patří Mgr. Tomáši Pochovi, Mgr. Ondřeji Šerému a Mgr. Pavlu Šafratovi za pomoc při řešení problémů s IVE. Poděkování též patří autorům programů Eclipse, GIMP, Vim, Openoffice a operačního systému Linux, které usnadnily dokončení této bakalářské práce.

Prohlašuji, že jsem svou bakalářskou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne

Josef Reidinger

Obsah

1	Úvod	6
1.1	Význam paměti	6
1.2	Cíle	7
2	Teoretické poznatky o fungování paměti a jejich využití v modelu	9
2.1	Model paměti	9
2.2	Význam spánku a uspořádání paměti	10
2.3	Zapomínání	10
2.4	Interference	11
2.5	Přehlcení novými vjemy	11
2.6	Hodnota vzpomínky	12
2.7	Shrnutí modelu paměti pro IVE	13
3	Porovnání modelu paměti s modely v jiných projektech	15
3.1	Enti	15
3.2	Potvůrka	16
4	O projektu IVE	18
4.1	Umístění paměti do architektury IVE	19
5	Upravení IVE pro potřebu paměti	22
5.1	Hlídání zmizení předmětu	22
5.2	Hledání nových předmětů	22
5.3	Zapamatování zajímavých věcí	23
5.4	Nové vyhodnocovací výrazy	24

5.5 Vylepšení volitelných předmětů	24
6 Testy	25
6.1 Test naplňování paměti	25
6.2 Test reakce na zmizení předmětu	27
6.3 Test vlivu více stejných předmětů	28
6.4 Test vlivu statické zajímavosti	31
6.5 Závěr testů	33
7 Závěr	35
7.1 Zkušenosti z vývoje	35
7.2 Možné budoucí rozšíření	36
8 Literatura	37
9 Seznam příloh	38

Název práce: Episodická paměť pro postavy z IVE

Autor: Josef Reidinger

Katedra: Kabinet software a výuky informatiky

Vedoucí bakalářské práce: Mgr. Cyril Brom

Email vedoucího: brom@ksvi.mff.cuni.cz

Abstrakt: Paměť je důležitou součástí každého člověka a na jejím fungování je závislý nejen rozvoj osobnosti, ale i každodenní chování. Je důležitá i pro věrohodné chování umělých bytostí. Tato práce má za cíl navrhnout model paměti, implementovat ho do projektu IVE a následně ověřit chování této paměti. Model má dvě oddělené části. Jedna slouží pro aktuálně spatřené nové předměty a druhá pro již déle známé předměty. Model má schopnost na požádání přesunout platné věci z první paměti do druhé, odstranit neplatné vzpomínky a ty, které se nevešly do kapacity paměti. Výběr odstraňovaných vzpomínek používá výpočet hodnoty vzpomínky na základě četnosti, posledního spatření, počtu podobných předmětů a statické zajímavosti objektu vzpomínky. Vzpomínky s nejmenší hodnotou se odstraní. Výsledný model se ukázal být vhodný za normálních podmínek a nepřesvědčivý při velkém množství předmětů stejného druhu.

Klíčová slova: episodická paměť, IVE, model paměti

Title: Episodic memory for project IVE

Author: Josef Reidinger

Department: Department of Software and Computer Science Education

Supervisor: Mgr. Cyril Brom

Supervisor's e-mail address : brom@ksvi.mff.cuni.cz

Abstract: Memory is an important part of each human and the progress of personality depends on its function. Everyday behavior also depends on it. So for artificial human it is also important for their authentic behavior. The aim of this work is to design, implement and test memory model for project IVE. The model has two separated parts. The first one is for perception of new objects and second is for already known objects. Model has an ability to move valid objects from the first part to second, remove invalid remembrance and remove overflowing objects. Removing of the overflowing objects uses calculating value of remembrance. This value depends on frequency sight, last sight, static value called interesting and count of same object's type in memory. Remembrance with lower value is removed. The final model is good in various environment and unsatisfactory when environment has a lot of same objects.

Keywords: episodic memory, IVE, memory model

1. Úvod

Základem každé hry je vytvořit prostředí, kde se má hráč pohybovat. Problémem bývá skutečně náhodný svět včetně jeho obyvatel. I když jsou zaznamenány pokusy vytvořit co nejvíce náhodné prostředí, tak každý vygenerovaný svět obsahuje stejné určité lokace. V současnosti se využívá především náhodné generování na úrovni rozmístění scén, ve kterých jsou kromě neživých předmětů i živé prvky (např. postavy) ovládané pouze počítačem tzv. NPC (non-player character). Umístit NPC do cizího prostředí je problematické, pokud tato postava nemá pouze stát nebo se bezcílně potulovat ve vymezeném prostoru.

Jedno z možných řešení existuje v projektu IVE (Intelligent Virtual Environment) [5], který tento problém odstraňuje pomocí teorie afordancí. Tato teorie rozpoznává předměty podle toho, k čemu slouží a ne jak vypadají. Uvažme kováře v cizím prostředí, který pro kování nebude shánět pouze kovářské kladivo, ale předmět, s kterým lze kovat (např. sekyru). Problémem projektu IVE je skutečnost, že postava musí všechny využitelné předměty vidět, aby s nimi mohla případně pracovat.

1.1 Význam paměti

Paměť je důležitou součástí každého člověka a na její činnosti je závislý nejen rozvoj osobnosti, ale i každodenní chování. Například ve filmu Memento je muž, který si nic nepamatuje od vraždy manželky a po jejím vrahovi pátrá za pomoci ukládání informací na papírky nebo tetováním na tělo. Stále si uchovává vzpomínky na dobu před vraždou. Jak by vypadal jeho život bez paměti už od dětství? Pokud by se jeden den spálil o plotýnku, nic by mu nebránilo se znovu spálit příští den, dokonce i bezprostředně po zvednutí ruky z plotýnky.

Přítomnost paměti je prokázána i u jednoduchých organismů [1]. Mezi pozorovatelné projevy paměti patří několik následujících schopností. Habituační (navykání) je schopnost vytvořit si návyk, kdy opakující se děj, který z počátku vyvolával

okamžitou reakci, začíná být postupně ignorován, pokud nemá pro organismus kladnou nebo zápornou odezvu. Opakem habituace je senzitivace (narůst citlivosti), kdy po prožitém podnětu dochází k zesílení reakce na podobné podněty. Například výstřel vyvolá okamžitou ostražitost, kdy i zapraskání větvičky vyvolá obdobnou reakci. Podmiňování je dalším snadno pozorovatelným jevem. Znáмым příkladem je Pavlovův pokus se psy. Ti si zapamatovali, že po zvuku zvonečku následuje jídlo. Pouhý zvuk zvonečku jim uvedl v činnost slinné žlázy.

1.2 Cíle

Cílem práce je navrhnout model paměti a implementovat ho do projektu IVE. Navržený model paměti by měl rozšířit možnosti IVE o znalost předmětů zaznamenaných v minulosti. Současně je záměrem ověřit si náročnost implementace navrženého modelu do již existujícího programu jak z hlediska práce programátora, tak z hlediska běhu programu. Pokud by se model ukázal dostatečně věrohodný a nenáročný, dal by se například využít v počítačových hrách, kde je interakce hráče a umělé bytosti. Ukázkou použití je vesnice, do které přijede hráč a hledá v ní určitý předmět. Počítačem řízené postavy (NPC) by se pohybovaly v prostředí vesnice dle svého zaměření (hospodský po hostinci, zahradník po zahradě atd.). Předmět by mohl být umístěn kdekoliv ve vesnici a pouze postavy, pohybující se v jeho okolí, by daly hráči správnou odpověď. Ještě ambicióznějším příkladem je hra, kde by se scény (budovy, obyvatelé, nepřátelé i předměty) generovaly náhodně. Poté by se spustila na tento svět simulace po určitou dobu. Hráč by se pak při vstupu do nové scény, kde by každý vesničan měl svoje vědomosti a zážitky, mohl pomocí jim položených vhodných otázek rychle zorientovat. Tím by se mohla prodloužit doba pobytu u hry, protože hráč by se při novém zahájení hry pohyboval v jiném světě. Další výhodou použití modelu paměti ve hře je, že hráči ke zjištění informací o prostředí, kde se pohybuje, stačí komunikovat s jeho obyvateli.

Podmínkou modelu paměti je, aby pracoval v reálném čase (realtime), a tudíž nebyl příliš náročný na počítač. Model by se měl zaměřit pouze na věci, které vnímá uživatel, případně jsou důležité pro věrohodnost simulace, a méně na věci, které simulují procesy

přímo v paměti. Potom by model paměti nemusel odpovídat vědeckým poznatkům o fungování paměti.

2. Teoretické poznatky o fungování paměti a jejich využití v modelu

Kapitola rozebírá některé znalosti z oblasti psychologie o fungování lidské paměti a jejich uplatnění v modelu paměti implementované do projektu IVE.

2.1 Model paměti

Různých modelů paměti existuje velké množství. Jeden z nejznámějších, nejvlivnějších a pro pochopení dobře popsáný je Atkinsonův a Shiffrinův model [1]. Podle něj se paměť dělí na sensorickou, krátkodobou a dlouhodobou.

Sensorická paměť umožňuje například sledovat film, protože dokáže navázat vjemy na sebe. Životnost informace v této paměti je dle pokusu s kolem a žhavými uhlíky desetina vteřiny [1].

Krátkodobá paměť, označovaná někdy jako pracovní, umožňuje například pochopit smysl vět, protože si pamatuje, co bylo na začátku věty.

Dlouhodobá paměť se ještě dělí na episodickou a sémantickou. Sémantická paměť dává postavě znalosti o světě a o vlastních dovednostech, u kterých není důležité, jak se získaly. Člověk umí číst, psát, poznat běžné předměty, aniž by si pamatoval, za jakých okolností tyto schopnosti získal. Jiné zdroje [6,7] definují sémantickou paměť pouze pro obecná fakta (např. která je sedmá planeta od slunce) a naučené činnosti (např. řezání pilou) přiřazují do procedurální paměti. Episodická paměť zajišťuje zapamatování si konkrétních událostí, u nichž si pamatuje i okolnosti, za kterých k nim došlo (oprava koňské podkovy před měsícem).

Použití v modelu IVE: Simulace je v IVE diskretní, proto se sensorická paměť nijak neprojeví. Data se předávají po celých částech a ne proudově, proto ani krátkodobá paměť není potřeba. Sémantická paměť i procedurální paměť se vzhledem k délce simulace neprojeví a je rovnou zadána designérem. Navržený model se bude zabývat především episodickou pamětí. Používá se krátkodobé paměti ve smyslu zapamatování si aktuálních nových prožitků a dlouhodobá paměť sloužící pro již dříve zapamatované

věci. Toto chování má zaručit, že postava si bude pamatovat jak nové věci, včetně jejich redukce (viz 2.5 *Přehlčení novými vjemy*), tak staré vzpomínky.

2.2 Význam spánku a uspořádání vzpomínek

O významu spánku se stále vedou dohady. Jisté je, že spánek je potřebný, protože se uchoval u všech živočišných i rostlinných druhů po celou dobu evoluce. Dohady o významu spánku se vedou přes biologické důvody (termoregulace, střádání metabolické energie), vývoj (zrání nervového systému) až po psychické důvody (narušení kognitivních funkcí) [4]. Existuje i názor, že spánek je důležitý, protože během něho dochází v mozku k uspořádání vzpomínek [4].

Použití v modelu IVE: Uspořádání vzpomínek v IVE znamená přesun všech vzpomínek z krátkodobé paměti do dlouhodobé. Přitom se také odstraní z dlouhodobé paměti všechny již neplatné vzpomínky. Model je schopen na požádání uspořádat vzpomínky. Do programu je vložena možnost uspořádat paměť během spánku.

2.3 Zapomínání

Často se stává, že lidé něco zapomenou. K tomuto jevu dochází třemi možnostmi [7]. První možností je, že mozek vyhodnotí informaci jako nepotřebnou nebo nepravdivou.

Druhou možností je postupné zapomínání vzpomínek, pokud se neobnovují. Naopak u často opakovaných vzpomínek dochází k jejich prohlubování.

Třetí možností jsou interference (více o interferencích níže).

Použití v modelu IVE: Do paměti se nepotřebné vzpomínky vůbec nedostávají, proto se nezapomínají. Nepravdivé informace se zapomínají během uspořádání vzpomínek. Postupné zapomínání je obsaženo v hodnotě vzpomínky (viz níže) obsahující frekvenci spatření. Ta se při delším nespatření předmětu snižuje a naopak při každém spatření se zvyšuje.

2.4 Interference

Interference je vzájemné ovlivňování vzpomínek a nastává v tom případě, pokud jsou v paměti podobné vzpomínky. Interferenci lze rozdělit na dva druhy – retroaktivní a proaktivní.

V *retroaktivní interferenci* nová vzpomínka přemazává starou. O tom, jestli nová vzpomínka oslabuje starou nebo ji pouze zastiňuje, se vedou spory.

Proaktivní interference naopak způsobuje, že se nečekaně vynoří stará zasunutá vzpomínka a je silnější než nová. V [1] je popisován případ, kdy autorovi v jeho oblíbené restauraci změnili značku piva. On si po roce objednal starou značku piva, přestože si celý rok objednával novou značku.

Použití v modelu IVE: Žádná z interferencí není použita v modelu, i když původně bylo zvažováno jejich zařazení. Aby paměť působila reálně, tak by muselo docházet k interferenci jen zřídka a přínos by vzhledem k vyšším nárokům na počítač nebyl efektivní.

2.5 Přehlcení novými vjemy

Při procvičování paměti se člověku ukáže určité množství předmětů a on si jich má zapamatovat co nejvíce. Zapamatovat si všechny předměty není jednoduché a dle mého pozorování se počet zapamatovaných předmětů pohybuje průměrně okolo deseti. Podobně hra pexeso závisí na zapamatování si počtu kombinací. Pokud člověk otočí druhou kartu z páru po méně otočených kartách, spíše si vzpomene na polohu první z páru. Stejně tak si člověk vědomě těžce vybaví vše, co viděl v pokoji, kde byl pouze jednou. Podobně je tomu po určitém časovém úseku, během kterého člověk prožije mnoho nových zážitků a zapamatuje si jen ty nejsilnější.

Použití v modelu IVE: Model se snaží tuto vlastnost simulovat krátkodobou pamětí, která je schopna zapamatovat si pouze omezené množství věcí a to ty nejzajímavější. „Zajímavost“ se posuzuje na základě vypočtené hodnoty (více níže). Protože jsou lidé různí a počet současně zapamatovaných informací je hodně individuální, což je vidět

například na hře pexeso, proto je možné nastavit velikost paměti.

2.6 Hodnota vzpomínky

V žádné mně dostupné literatuře není popsán přesný vzorek či postup pro určení vzpomínky, která má vytlačit jinou. Určitě na tom má podíl četnost opakování stejné vzpomínky a také poslední obnovení vzpomínky (více v 2.3 *Zapamatování*). Důležité také je, jak na nás objekt zájmu emočně působí a za jakých okolností si jej zapamatujeme. Příklad dokumentující tuto vlastnost lidské paměti je v [1].

Použití v modelu IVE: Četnost opakování a poslední spatření předmětu je použito k výpočtu zajímavosti. Zohlednění emočního působení v modelu paměti by bylo poměrně náročné, proto je využita pouze vlastnost předmětů pojmenovaná jako interesting („zajímavost“) a vyjadřuje, jak předmět vnímají postavy pohybující se kolem něho. Pro optimalizaci je přidán parametr, který udává počet podobných věcí už obsažených v paměti, aby často se opakující předmět nezahltil celou paměť. Četnost spatření se snižuje za každou hodinu, co nebyl objekt spatřen, a zvyšuje se při každém novém spatření. Hodnota četnosti má nastavené své meze, aby předmět používaný postavou celý den neměl nepřiměřeně vysokou hodnotu a naopak zajímavý předmět měl přiměřeně nízkou hodnotu, pokud nebyl spatřen delší dobu. Vzoreček na výpočet hodnoty vzpomínky je ponechán otevřený a je možné zkoušet různé varianty závislosti výše popsanych parametrů na sobě nebo některý z nich vynechat. Také je možné použít různé vzorce pro dlouhodobou a krátkodobou paměť. Základní varianta je níže uvedený vzorec pro obě paměti.

$$v = \frac{(a+b)}{(c+d)}$$

v – výsledná vypočtená hodnota vzpomínky

a – četnost spatření předmětu

b – hodnota statické zajímavosti předmětu

c – čas uplynulý od posledního spatření (v hodinách)

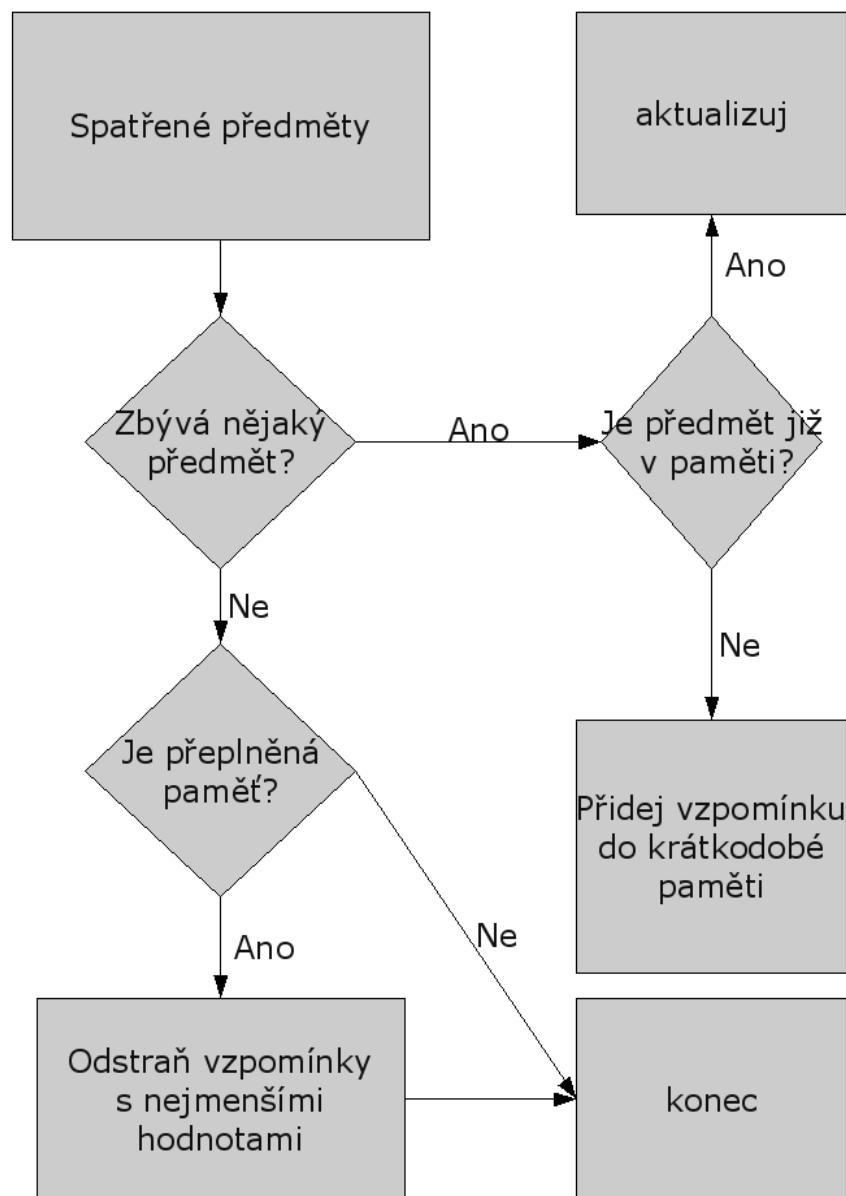
d – počet stejných předmětů v krátkodobé nebo dlouhodobé paměti (záleží na umístění vzpomínky)

2.7 Shrnutí modelu paměti pro IVE

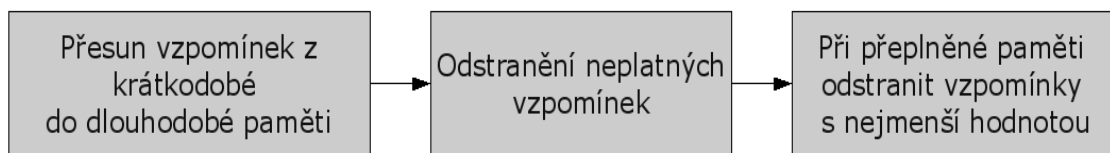
Model simuluje pouze sémantickou paměť rozdělenou na dvě samostatné části – krátkodobou a dlouhodobou. Krátkodobá paměť slouží pro nově spatřené předměty (obrázek 1). Dlouhodobá paměť obsahuje již zapamatované vzpomínky. U krátkodobé paměti se očekávají častější změny obsahu, proto je na ně optimalizována. Složitost přidání nebo odebrání vzpomínky z této paměti je lineárně závislá na její velikosti ($O(n)$). Dlouhodobá paměť slouží pro stálější vzpomínky, proto je optimalizována na rychlý přístup ke vzpomínkám. Složitost vyhledání vzpomínky v této paměti je konstantní ($O(1)$).

Nové vzpomínky se přidávají do krátkodobé paměti při vykonávání činností a na požadavek uživatele. Postava má omezenou krátkodobou paměť, proto si při přeplnění novými vzpomínkami zapamatovala jen ty s největší hodnotou. V rámci úspory paměťových nároků programu si paměť pamatuje pouze poslední zaznamenané údaje o předmětu.

Uspořádání paměti se provádí na požádání a při tomto procesu proběhne odstranění neplatných vzpomínek, přesunutí vzpomínek z krátkodobé paměti do dlouhodobé a odstranění vzpomínek s nejmenší hodnotou, pokud je překročena kapacita dlouhodobé paměti (obrázek 2). Složitost tohoto uspořádání je lineárně závislá na součtu velikostí obou pamětí ($O(n+m)$). Vzorec pro hodnotu vzpomínky závisí na četnosti spatření, době od posledního spatření, zajímavosti předmětu a počtu stejných předmětů v paměti. Vzorec může být uživatelsky zadán a může být různý pro dlouhodobou a krátkodobou paměť.



Obrázek 1 – Diagram činností při spatření nových předmětů



Obrázek 2 – Sekvenční diagram vykonávaných činností při uspořádání paměti

3. Porovnání modelu paměti s modely v jiných projektech

Kapitola porovnává model paměti implementovaný do IVE s modely použitými v jiných projektech.

3.1 Enti

Enti jsou autonomní agenti, kteří svým chováním napodobují lidi - starají se o svůj virtuální svět a snaží se v něm přežít formou uspokojování svých základních životních potřeb. Enti se pohybují v diskretním světě. Prostor je tvořen dvourozměrnou čtvercovou sítí. Mezi počítačem řízenými enty se pohybují i enti řízení lidmi [8].

Reprezentace paměti

Data v paměti, která je u entů tvořena databází, jsou ukládána v podobě „vět“ s pevně daným počtem, pořadím a významem parametrů. Toto má za cíl usnadnit si zapamatování vět vyřčených jinými enty, odpovídání na otázky a porovnávání relevantnosti záznamů. Data se ukládají v co nejjednodušších informacích. Složitější informace se musí vytvářet z jednoduchých informací pomocí tranzitivity nebo skládání. To má za následek oddělení vlastností předmětu od sebe a tím následně horší efektivitu vyhledávání určitých informací.

Ukládaná data

V paměti entů se vyskytují tři typy informací. Prvním typem jsou neměnná data, jako jsou ISA vztahy (něco je něco), architektura domu a základní informace o předmětech (např. co se čím čistí). Druhým typem jsou data, která se v průběhu simulace mění, jako je umístění a stav předmětu nebo pocity (např. hlad) enta. Počet těchto měnících se dat je omezen na cca padesát. Posledním zapamatovávaným typem dat jsou věty ostatních entů. U měnících se vlastností se zapamatuje pouze poslední stav a nelze zjistit předchozí stavy.

Porovnání paměti entů a v IVE

Paměť v IVE si ukládá na rozdíl od entů pouze data, která se mění, ostatní jsou uchována přímo v programu. Pamatování vět jiných postav není implementováno, protože postavy spolu v IVE nekomunikují a ani jinou formou si informace nepředávají. IVE obsahuje na rozdíl od entů dělení paměti na krátkodobou a dlouhodobou, což ovlivňuje chování bytostí u nových vjemů. U obou projektů se posuzuje věrohodnost, pouze v IVE je možné si definovat vlastní výpočet hodnoty pro věrohodnost (jak u krátkodobé, tak u dlouhodobé paměti). Obě paměti uchovávají pouze nejnovější informace. Shodné u obou projektů je, že vzpomínka se buď pamatuje nebo ne. Neexistuje žádný mezistupeň, ve kterém by se uchovala pouze část vzpomínky.

3.2 Potvůrka

Cílem projektu bylo vytvořit takovou bytost, která ve virtuálním světě vykonává jednoduché úkoly a pamatuje si, co podstatného se děje kolem ní. Bytost umí odpovídat na dotazy typu: "co jsi dělal v ...", "kde leží předmět...", "kdy jsi dělal ..." apod. v rozumném čase a na nejčastější dotazy v konstantním čase. Svět, bytost i její paměť jsou navrženy podle teorie afordancí [5,9].

Paměť v Potvůrce

V paměti je reprezentovaná sada všech procesů. Do paměti se zaznamenávají vztahy mezi procesy a zdroji (tj. zdroj je zaznamenán do paměti v závislosti na procesu, k jehož vykonání byl použit). Paměť má omezenou kapacitu a staré věci se zapomínají podle principu fronty. Do paměti se ukládají vztahy procesy a zdroji, údaje o procesech a explicitně zadané údaje (například údaje zadané uživatelem). U každého vztahu (proces – zdroj) se ukládá čas vzniku vztahu a místo zdroje před procesem a po něm.

Porovnání paměti v Potvůrce a v IVE

Paměť v Potvůrce je navržena k jinému účelu. Potvůrka je schopna odpovídat na otázky, co kdy dělala, kdy byl jaký předmět na jakém místě nebo kde je předmět teď.

IVE má složitější systém pro odstraňování starých vzpomínek, což je dáno i větší rozsáhlostí a rozmanitostí prostředí, kdy lineární zapomínání bez vyhodnocení důležitosti věcí by mohlo vést k zapomínání všech věcí zapamatovaných před příchodem do velmi zaplněné místnosti. V Potvůrce je nutná paměť pouze pro jednu bytost, proto je možné mít v paměti více informací. Nezajímavé předměty se v Potvůrce filtrují před zapamatováním. IVE naopak musí z důvodu využití paměti pro více bytostí jejich obsah redukovat. Proto se neuchovává starší umístění předmětů, ani vykonávané činnosti.

4. O projektu IVE

Kapitola popisuje projekt IVE a způsob implementace modelu paměti do něho.

Program IVE je výsledkem studentského softwarového projektu na MFF UK. Projekt se zaměřuje na snadnou rozšiřitelnost a efektivní simulaci rozsáhlých světů.

Rozšiřitelnost je řešena pomocí vylepšených „chytrých“ předmětů. Chytré předměty je metoda, která je založena na teorii afordancí britského psychologa J. J. Gibsona. Principem je vložení inteligence světa do předmětů, které samy říkají postavě, co se s nimi má provádět. Postava se snaží uspokojit svoji potřeby, rozhlídne se po okolí, vybere si předmět, který nabízí uspokojení své potřeby, a dál se nechá vést tímto předmětem. IVE tuto metodu dále rozšiřuje o chytré procesy, chytré rady a systém géníů.

Chytré procesy usnadňují akce, kde je potřeba zapojit současně více předmětů. Proces si sám shání zdroje, které potřebuje k provedení. Tyto procesy jsou hierarchicky uspořádané a každá postava má svůj životní proces, který uskutečňuje. Horník může mít svůj denní proces nastaven například jako - pracuj v dole, jdi do hospody a spi. Každý proces má svůj spouštěč a kontext. Spouštěč říká, jestli je možné začít tento proces. Například u horníka není vhodné jít do hospody v pracovní době, a proto má svůj spouštěč nastaven na stav „v pořádku“ pouze mimo pracovní dobu. Kontext naopak určuje, kdy se má proces ukončit. Například u procesu „dojít k posteli“ kontext řekne, že se má proces ukončit, ve chvíli, kdy je postava u postele.

Chytré rady určují vhodnost předmětu pro daný proces. Účelem těchto rad je vybrat vhodné spojení různých zdrojů. Například slabý kovář si nevezme na kování nejtěžší kladivo, ale přiměřeně těžké své síle. Proces vybere předměty, které má k dispozici, a chytrá rada řekne, jak je který vhodný. Tato hodnota je pouze doporučující a proces se může uskutečnit i pokud je tato hodnota velmi nízká. Rady mají také svůj kontext, který se v průběhu procesu mění. Například zmiňovaný kovář během kování poškodí kladivo, kontext se sníží a on zváží jeho výměnu.

System géníů slouží pro správné chování v určitých místech a v určitých situacích.

Génius je samostatná inteligence, která umí v daném prostředí a za dané situace převzít kontrolu nad jednou či více postavami. Například v hospodě může být génius na roznášení piva, který řídí objednávání a nošení piva mezi hosty a číšníkem.

Efektivita simulace je zajištěna hlavně díky použití úrovně detailů (LOD) pro různé lokace. Plně simulovány jsou lokace, na které je zaměřena uživatelova pozornost. V ostatních lokacích probíhají pouze zjednodušené procesy s výjimkou procesů souvisejících s dějem, které jsou také plně simulovány.

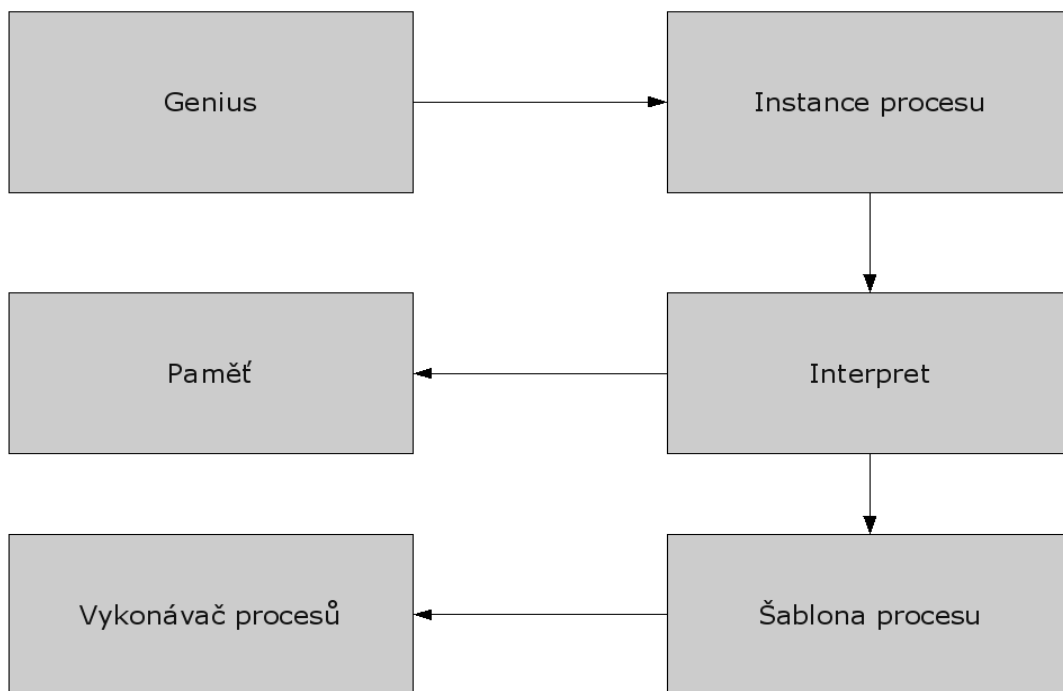
4.1 Umístění paměti do architektury IVE

Přidávání vzpomínek do paměti probíhá dvěma způsoby. Prvním (běžným) způsobem si postava zapamatuje předměty, s kterými pracuje. Druhým způsobem, který využívá spouštěč a kontext procesu, je hledání speciálních zajímavých předmětů. Tento způsob je více popsán v podkapitolách 5.3 *Zapamatování zajímavých věcí* a 5.4 *Nové vyhodnocovací výrazy*. První způsob je centrálně umístěn v interpretu procesů, aby zachytil každý proces a nemusel se vkládat do každého procesu nebo šablony zvlášť (viz obrázek 3). Z procesu se vezmou všechny vůdčí postavy a každé postavě, která má paměť, jsou předány všechny zdroje procesu. Paměť si vyfiltruje svoji postavu a všechny zdroje týkající se míst, protože nejsou potřeba.

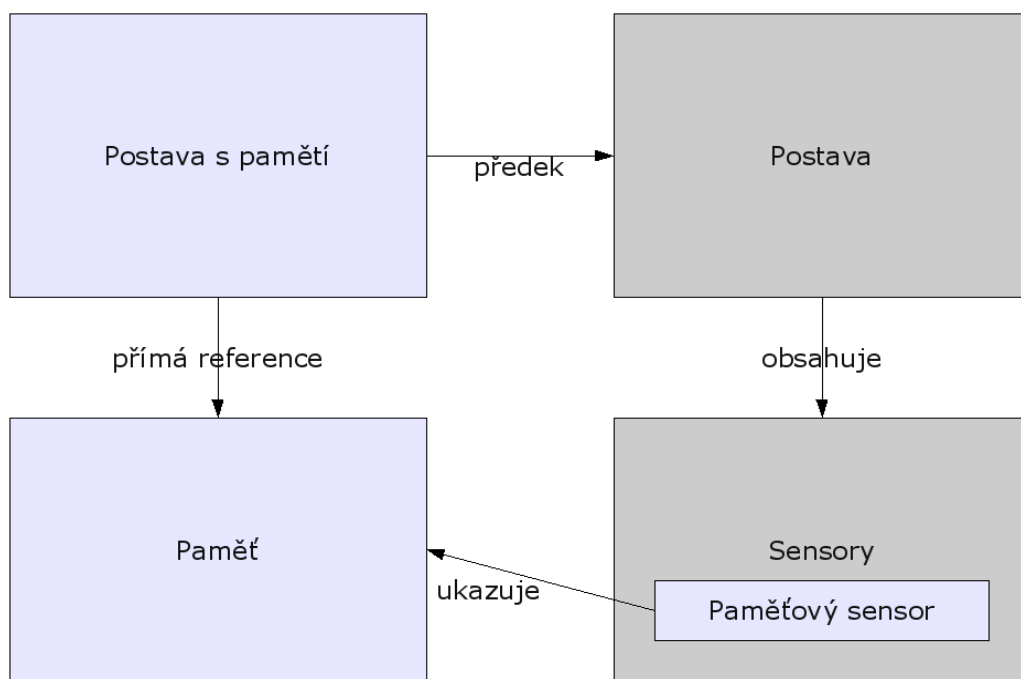
Využití paměti je možné dvěma způsoby (obrázek 4). Prvním způsobem je umístění paměti jako sensoru do seznamu sensorů obyčejné postavy. Tento přístup je převážně využíván pro vyhodnocovací výrazy (algoritmus na výpočet hodnoty pro spouštěč, kontext nebo chytrou radu). Tento způsob je používán, protože vyhodnocovací výrazy mají hierarchickou strukturu a předávají si navzájem svoje sensory, které využívají ke svému vyhodnocení. Tím i původní výrazy z IVE mohou předat paměť v seznamu sensorů svým podvýrazu. Naopak ve chvíli, kdy vyhodnocování dojde až k výrazu, který potřebuje paměť, tak je v seznamu sensorů k dispozici. Toto řešení umožňuje, aby se v simulaci vyskytovaly vedle sebe postavy s pamětí a bez ní. Druhým způsobem je postava s přímým odkazem na paměť. Tento způsob se využívá při naplňování paměti a je výhodný, protože se nemusí procházet celý seznam sensorů, ale paměť je rovnou k dispozici.

Paměť se používá i při naplňování zdrojů procesu. Při vytváření nové postavy se osobnímu géniovi postavy změní rozhodovač. Ten vyhledává a následně vybere, čím budou naplněny procesy, které postava vykonává. Původní rozhodovač sloty procesu (jednotlivé podmiňující objekty pro danou činnost) nejprve naplní tím, co už má postava u sebe. Potom zkusí doplnit z předmětů, které jsou již obsaženy ve slotech. Nakonec zkusí naplnit slot předměty, které vidí. Nový rozhodovač navíc, pokud nic vhodného nevidí, zkusí paměť postavy. Toto řešení umožňuje logický výběr, kdy postava teprve ve chvíli, kdy nevidí žádný vhodný předmět, začne pátrat v paměti.

Paměť nefunguje při částečně rozbaleném světě nebo při změně rozsahu rozbalení během simulace. Při částečně rozbaleném světě postava podvádí své rozdělené oční senzory a kouká i do sousedních místností, což znemožňuje hledání zajímavých věcí a vede k zacyklení programu. Postava neví, ve které je místnosti, neboť vidí všechny místnosti i s věcmi uvnitř. Problém nastává i u postavy, která není detailně simulována, a dostane se do detailně simulovaného místa. Ta nebude znát detailní informace o navštívených vedlejších místnostech (nebyly simulovány). To vede například k tomu, že postava uspokojována pitím a jídlem v hospodě, kde probíhal proces ve zjednodušené verzi, si bude pamatovat pouze hospodu a už ne, co jedla či pila. Podobně při návštěvě místa, kde se neodehrávala simulace, bude mít lokální postava zpočátku prázdnou paměť (žádné vzpomínky).



Obrázek 3 – diagram posloupnosti volání metod tříd při vykonávání procesu a napojení paměti (šedé rámečky třídy, šipky volání metod)



Obrázek 4 - umístění a přístup k paměti (šedé rámečky třídy původní aplikace, modré rámečky nově přidané třídy, šipky vztahy mezi třídami)

5. Upravení IVE pro potřeby paměti

Kapitola popisuje změny provedené v IVE, které se netýkají přímo paměti, ale jsou její funkčnost důležité.

5.1 Hlídaní zmizení předmětu

Pro aktualizaci předmětů v paměti se využívá systému aktualizace phantomů (virtuální kopie předmětů), který je již v IVE k dispozici. Každá změna původního předmětu se oznámí všem posluchačům, kteří se přihlásili k odebrání těchto aktualizací (podrobnosti v [5]). Zde nastává problém, neboť tento systém aktualizací nepočítá s tím, že předmět nemusí být stále vidět, může se někam přesunout a postava uvidí pouze prázdné místo. Proto byla doplněna do manažeru smyslů funkcionalita, která informuje, že postava vidí konkrétní místo a lze poté lehce zkontrolovat, jestli je na daném místě předmět skutečně umístěn.

5.2 Hledání nových předmětů

Původní prostředí IVE počítalo s tím, že vše potřebné, postava okamžitě pomocí smyslů vidí. Tento způsob se příliš nehodí pro paměť. Postava by nepotřebovala paměť, kdyby všechny předměty ihned viděla. Okamžitě po spuštění simulace je paměť prázdná a nemůže tedy nabídnout vhodné vzpomínky pro plnění úkolů postavy. Při běhu simulace se také může stát, že v paměti není vhodný předmět pro provedení činnosti a je nutné umožnit postavě pátrat v okolí po předmětech, které se jí hodí. Protože pohyb postavy v neznámém prostředí nebyl úkolem této práce, je implementováno pouze jednoduché pátrání, kdy si postava náhodně vybere lokaci a jde do ní. Aby se postava mohla takto pohybovat, jsou rozděleny sensory postavy na sensor lokální, který vidí věci pouze v aktuální lokaci (kde se nachází) a na sensor prostředí, kdy postava vidí všechny lokace. Při případném implementování pohybu v neznámém prostředí by se dala využít i paměť. Postava by pak neviděla všechny lokace, ale znala by místa, kde se dá přejít do neznámé lokace. Již navštívené lokace by byly uloženy v paměti.

5.3 Zapamatování zajímavých věcí

Potřebnou funkcionalitou také je, aby si postavy mohly zapamatovat objekt se speciální vlastností – zajímavost. O tom, který předmět má tuto vlastnost, rozhoduje návrhář předmětů. Problémem je, jak toto integrovat do IVE, když zajímavá věc nemusí být v místnosti, nebo může být v místnosti, ale není v paměti, protože nebyla vyhodnocena jako dost zajímavá ve vazbě na kapacitu paměti. Tímto by mohl vzniknout nekonečný cyklus z pokusů ji přidat do paměti.

Prvním pokusem o řešení byl prostý čekací proces, který začne ve chvíli, kdy je nějaký zajímavý předmět v místnosti a není v paměti. Na předmět byl použit volitelný slot, u kterého se kontrolovalo, jestli je naplněný. Ukázalo se, že IVE volitelné sloty samo nevyplňuje. Při pokusu dodat tuto funkcionalitu, přestaly fungovat některé procesy z původního demovsvěta. Proto toto řešení nebylo v rámci zpětné kompatibility použito.

Dalším pokusem o řešení bylo využít existujícího výrazu doisee, který zkusil najít objekt splňující dané požadavky a doplnil ho do volitelného slotu. Problémem řešení je, že doisee uvidělo jeden objekt a ten doplnilo. Pokud byl objekt stále vidět, tak nemělo o další objekty zájem, dokud se nezměnila scéna (přešlo se do jiné místnosti). To vedlo k tomu, že si postava všimla pouze jednoho předmětu, i když v místnosti bylo více zajímavých předmětů. To nelze pokládat za vhodné řešení.

Tento problém měl napravit výrazový podstrom, umožňující vyjádřit podmínku na spatřený objekt. Doisee tuto nedokumentovanou vlastnost již mělo. Tento postup nevedl k požadovanému cíli. Po naplnění slotu prvním spatřeným objektem proběhl proces zapamatování a výraz doisee pouze hlídal, zda je objekt vidět, ale neověřoval, zda stále splňuje výrazový podstrom. Nesnažil se o naplnění slotu jiným objektem, který by si mohl zapamatovat. Jediný rozdíl ve výsledku při použití podstromu byl, že vybraný objekt nebyl v paměti.

Konečné řešení využívá přidanou vlastnost paměti, která si pamatuje poslední

zkoumanou oblast na zajímavé předměty. Pokud se postava dostane do jiné místnosti, tak se pokusí zapamatovat si všechny zajímavé věci a aktualizuje poslední prohledanou místnost. Detailně je toto řešení popsáno v následujícím odstavci.

5.4 Nové vyhodnocovací výrazy

K výrazům, které je možné použít pro vyhodnocení spuštění nebo pokračování procesu (více v kapitole 4. *O projektu IVE*, v odstavci o chytrých procesech), jsou přidány *isinmemory*, *seenotinmemory* a *fillnotinmemory*. *Isinmemory* slouží pro ověření, zda je předmět stále v paměti. V případě, že je předmět odstraněn nebo přidán, tak je o tom výraz informován a aktualizuje se. Druhý a třetí se používají spolu. *Seenotinmemory* zkoumá, jestli jsou v místnosti vidět zajímavé předměty a místnost nebyla prohledána jako poslední. *Fillnotinmemory* doplní zajímavé předměty do paměti a zajistí, že *seenotinmemory* se aktualizuje.

5.5 Vylepšení volitelných předmětů

Pro snadnější přidávání volitelných předmětů (předmětů, které se během simulace mohou přidávat a odebírat) je doplněna možnost snadněji si volit přímo v XML, které předměty jsou volitelné. Do IVE je přidán dialog pro výběr předmětů, které se objeví nebo zmizí. Zde bylo potřeba zlepšit použití tohoto mechanismu pro návrháře lokace. Původní řešení v IVE používalo natvrdo zadaný kód do javy, což se pro více předmětů ukázalo nevhodné a pro správu těchto předmětů téměř nepoužitelné. Tato změna se netýká přímo paměti, ale ukázala se být důležitá pro testování zmizení předmětů.

6. Testy

V této kapitole jsou popsány testy provedené na paměti implementované do IVE. Testuje se hlavně chování v extrémních situacích.

6.1 Test naplňování paměti

Cíl testu

Cílem testu je ověřit chování paměti při úvodním naplňování. Při uvažovaném použití v počítačové hře je to moment po vygenerování světa, kde by postavy měly určitou dobu žít před příchodem hráče, aby znaly své okolí a dokázaly se po něm pohybovat a říkat o něm smysluplné informace.

Popis testu

Testované prostředí má simulovat jednoduchou postavu, zde konkrétně farmáře, ve hře. V levé lokaci se nachází květina (simulující pole), u které farmář pracuje přes den od jedenácti dopoledne do dvou hodin ráno. Poté hodinu prochází přes lokace až dojde do správné lokace, kde je jeho postel. Zde spí od tří ráno do deseti hodin dopoledne. V prostřední místnosti se nachází padesát náhodně rozmístěných truhel, které jsou zařazeny mezi zajímavé předměty. Postava si jich během průchodu všímá. Kromě těchto truhel se do paměti dostane i kytky a postel, protože postava je používá ke svým činnostem. Testuje se celkem pět velikostí krátkodobé paměti – 2, 5, 10, 20, 50. Dlouhodobá paměť je dostatečně velká na uchování všech předmětů. Synchronizace krátkodobé a dlouhodobé paměti probíhá během spánku.

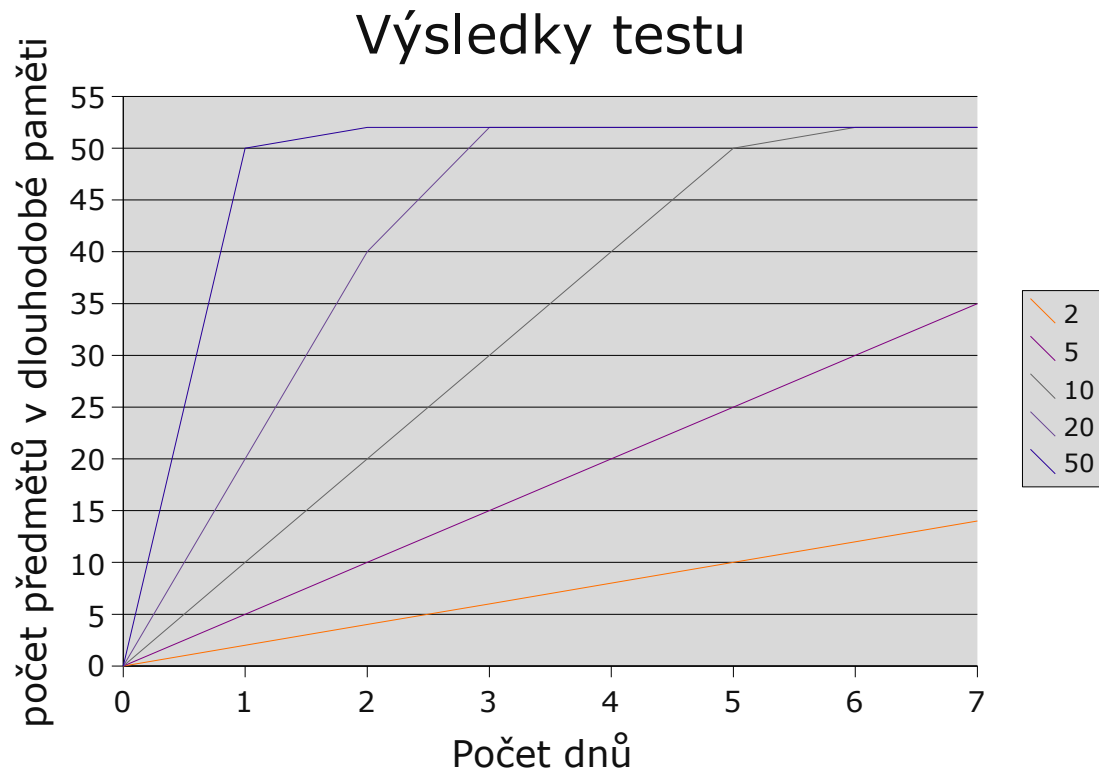
Měření počtu předmětů v obou pamětech probíhá vždy těsně po půlnoci po dobu sedmi dnů. Pro účely testu byla vytvořena třída *Test1*, která provede celé měření pro všech pět pamětí a výsledek uloží v aktuálním adresáři do souboru *test1_result.txt*. Test používá scénáře *farmer1-5.xml*.

Výsledek a zhodnocení testu

den\vkp*	2	5	10	20	50
0	0	0	0	0	0
1	2	5	10	20	50
2	4	10	20	40	52
3	6	15	30	52	52
4	8	20	40	52	52
5	10	25	50	52	52
6	12	30	52	52	52
7	14	35	52	52	52

Tabulka 1 – výsledek testu naplňování paměti

* vkp - velikost krátkodobé paměti



Obrázek 5 – graf naplňování paměti

Z výsledků testu jednoznačně vyplývá, že rychlost zapamatování si úvodního stavu je lineárně závislá na velikosti krátkodobé paměti a staré vzpomínky v dlouhodobé paměti vůbec neovlivňují zapamatování si nových vzpomínek. Této skutečnosti by se dalo využít po vygenerování světa pro zkrácení času nutného na simulaci počátečního stavu, kdy by se dočasně zvýšila hodnota krátkodobé paměti a pro simulaci s hráčem, kdy by se zase snížila na původní velikost.

6.2 Test reakce na zmizení předmětů

Cíl testu

Cílem testu je ověřit si reakci paměti na zmizení předmětů ze svého místa. Toto chování není ošetřeno pomocí pasivních aktualizací, více v kapitole 5.1 *Hlídaní zmizení předmětů*.

Popis testu

Testované prostředí simuluje hraničáře, který kontroluje svůj úsek hranice. V simulaci je domovská lokace, kde hraničář spí, a dalších devět propojených lokací, kde hraničář prochází a hlídá hranici. Na začátku nejsou předměty v simulaci. Předměty jsou volitelné a aktivovat je lze běžnými prostředky popsány v uživatelské dokumentaci.

Pro účely tohoto testu byla vytvořena třída *Test2*, která měří během půlnoci počet předmětů v obou částech paměti. O půlnoci se také změní stav předmětů, buď se přidají nebo odeberou ze simulace. Měření se zapisuje do *test2_result.txt*. K měření se používá scénář *hranicar.xml*. Pokud se měří ručně přes spuštěnou aplikaci, lze za běhu i sledovat, jak se předměty pouze zneplatní a neodstraní z paměti. Odstranění proběhne až během synchronizace paměti během spánku. Při této simulaci lze pozorovat, jak se zmizením předmětu stává vzpomínka na něj neplatnou, ale přesto je stále v paměti. Pokud tento předmět postava znovu spatří, mezi spatřením a zneplatněním neproběhla synchronizace, tak se původní vzpomínka stane platná a nemusí se znovu přidávat.

Výsledek a zhodnocení testu

den	počet předmětů v kp*	počet předmětů v dp ⁺
0	0	0
1	12	12
2	0	1
3	12	13
4	0	1
5	12	13
6	0	1
7	12	13
8	0	1
9	12	13
10	0	1

Tabulka 2 – test reakce na zmizení předmětů

**kp - krátkodobá paměť*

+ dp – dlouhodobá paměť

Test ukázal, že dlouhodobá paměť reaguje podle předpokladů na zmizení předmětů. Jediný nemizící předmět je postel, kterou postava používá na spaní. Pro hru, pokud by měla synchronizaci také při spánku, by to znamenalo, že je postava schopná v daném dni říci o zmizení předmětu, ale příští den by na něj zapomněla a nic o něm nevěděla.

6.3 Test vlivu více stejných předmětů

Cíl testu

Cílem testu je zjistit vliv více stejných předmětů na hodnotu vzpomínky a reálné chování mechanismu ochrany paměti před zahlcením stejnými předměty. Dalším cílem je demonstrovat možnosti změny výpočtu hodnoty vzpomínky a vliv na paměť.

Popis testu

Prostředí je stejné jako u předchozího testu. Předměty již nejsou volitelné a k truhlám přibyly ještě meče, jako druhý typ předmětů. Je přítomno padesát truhel a deset mečů v prvním testu a sto padesát truhel a třicet mečů v druhém testu. V obou případech je velikost dlouhodobé paměti omezena na patnáct, aby se uplatnily rozhodovací pravidla o tom, které předměty vyřadit z paměti. Druhý test využívá speciálně definovaný vzorec, který dává větší váhu četnosti stejných předmětů v paměti.

Pro účely testu byla vytvořena spustitelná třída *Test3*, která provádí měření o půlnoci. Měří počet mečů a truhel v každé z pamětí. Pro spuštění třídy je nutné do cesty pro hledání tříd přidat i *memory.jar* z adresáře *scenare*, který obsahuje upravenou formuli na výpočet hodnoty. Testy se provádí na scénářích *hranicar2-3.xml*.

Výsledek a zhodnocení testu

den	a	b	c	d	e	f	g	h	I	j	k	l
0	0	0	0	0	0	0	0	0	0	0	0	0
1	8	3	0,73	5	7	0,42	1	10	0,09	1	11	0,08
2	11	0	1	5	10	0,33	1	11	0,08	2	12	0,14
3	11	0	1	5	10	0,33	2	9	0,18	3	12	0,2
4	11	0	1	5	10	0,33	2	10	0,17	5	9	0,36
5	11	0	1	5	10	0,33	2	9	0,18	7	8	0,47
6	11	0	1	5	10	0,33	1	11	0,08	9	5	0,64
7	11	0	1	5	10	0,33	0	11	0	10	5	0,67
8	11	0	1	5	10	0,33	1	10	0,09	10	5	0,67
9	11	0	1	5	10	0,33	1	10	0,09	11	4	0,73
10	11	0	1	5	10	0,33	0	11	0	12	3	0,8

Tabulka 3 – test více stejných předmětů

Vysvětlivky k tabulce a grafu

a,g – počet truhel v *kp* v testu 1, 2

b,h – počet mečů v *kp* v testu 1, 2

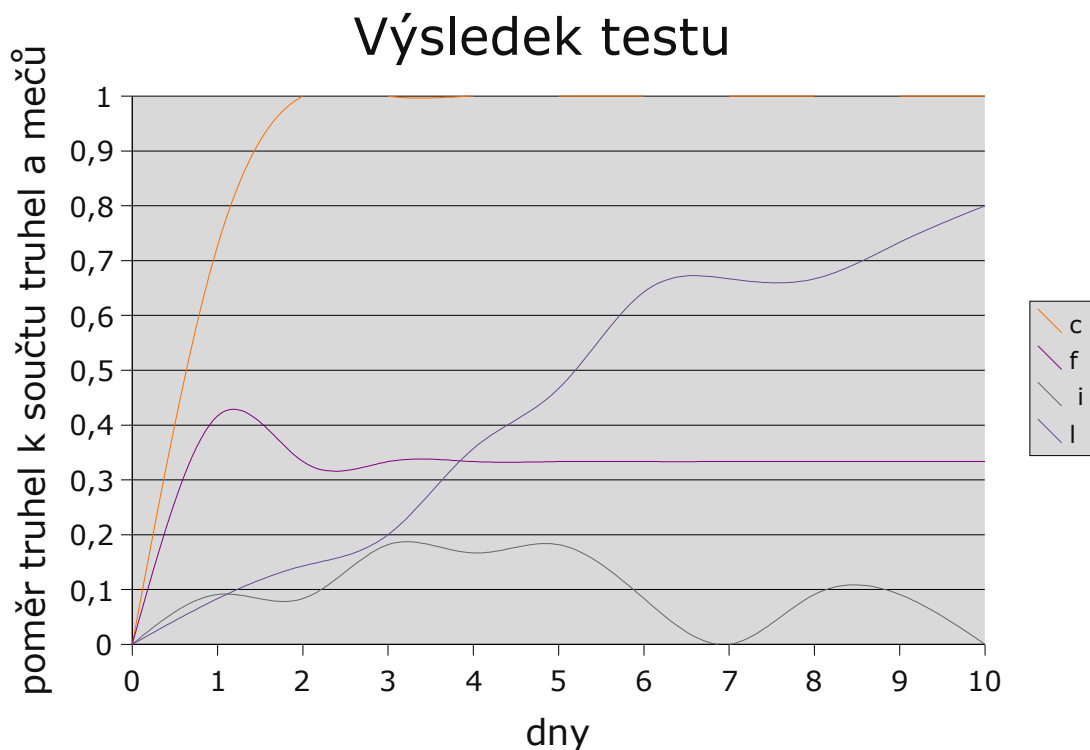
c,f,I,l – poměr truhel k součtu truhel a mečů v testu 1 v *kp* a *dp* a testu 2 v *kp* a *dp*

d,j – počet truhel v *dp* v testu 1, 2

e, k – počet mečů v dp v testu 1, 2

kp – krátkodobá paměť

dp – dlouhodobá paměť



Obrázek 6 – graf poměru předmětů v paměti v závislosti na čase

První test ukázal, že se předměty v menšině rychle dostaly do paměti. Pomohla už krátkodobá paměť, kdy při vstupu do místnosti bylo k vidění více truhel než mečů a tak truhly dostaly výhodu. Tím, že se hodnota počítá pouze jednou a ne při každém jednotlivém odstranění předmětu, dochází k tomu, že v krátkodobé i dlouhodobé paměti není rovnováha předmětů. Více je těch, co byli před redukcí v menšině. Zajímavá je skutečnost, kterou ukázal tento test, že postel, jediná svého druhu, byla postavou viděna pouze před spaním a tím neměla dostatečnou hodnotu pro přidání do paměti.

Druhý test ukázal, že, při velkém počtu předmětů a ještě výrazně změněným vzorkem na výpočet hodnoty vzpomínky, ovládly krátkodobou paměť meče. Při každém vstupu do místnosti bylo spatřeno průměrně sedmnáct truhel a tři meče

a tím byla hodnota truhel výrazně nižší. V dlouhodobé paměti se trend obrací. Z krátkodobé paměti nepřichází příliš velké množství truhel a tím mají v dlouhodobé paměti větší hodnotu a postupně v ní získávají převahu. Zajímavé je, že pro větší důležitost počtu stejných objektů se občas dostane do paměti i postel. Tento případ je dán vzorečkem, kdy jsou použity pouze dva druhy předmětů, které svým množstvím přesahují i dlouhodobou paměť, zároveň slouží jako ukázka, pro které případy se tato paměť nehodí. Pro tento případ by se více hodila paměť, která by dokázala shlukovat velké množství stejných předmětů do jednoho a jen si pamatovala, v které místnosti tyto předměty byly.

6.4 Test vlivu statické zajímavosti

Cíl testu

Cílem testu je zjistit chování paměti u předmětů s různou statickou zajímavostí a současně fungování ochrany před přeplněním stejnými předměty, které mají větší statickou zajímavost než ostatní.

Popis testu

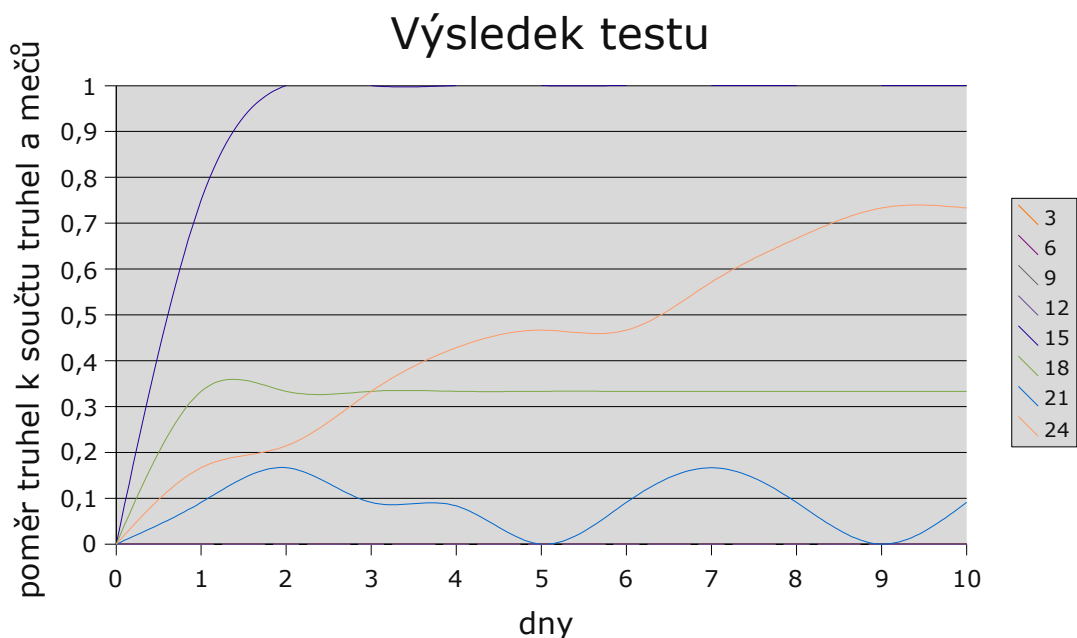
Prostředí je stejné jako u předchozího testu. Předměty jsou opět truhly a meče. Truhly mají přednastavenou zajímavost jedna a meče mají zajímavost sto. Dlouhodobá paměť je stále snižena na patnáct. Čtyři různá nastavení testu se liší v počtu předmětů a v použití upraveného výpočtu z předchozího testu. První nastavení má padesát mečů i truhel a přednastavené ohodnocování vzpomínek. Druhé nastavení má k dispozici upravený způsob výpočtu. Třetí používá také upravený způsob vyhodnocování a má k dispozici pouze deset truhel a mečů. Čtvrté nastavení je stejné jako druhé, jen je zajímavost mečů snížena na pět.

Pro účely testu byla vytvořena spustitelná třída *Test4*, která provádí měření o půlnoci. Měří počet mečů a truhel v každé z pamětí. Pro spuštění třídy je nutné do cesty pro hledání tříd přidat i *memory.jar* z adresáře *scenare*, který obsahuje upravenou

formuli na výpočet hodnoty vzpomínky. Testy se provádí na scénářích *hranicar4-7.xml*.

dny	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	12	0	0	12	0	0	11	0	0	12	0	6	2	0,75	4	8	0,33	1	10	0,09	2	10	0,17
2	0	12	0	0	15	0	0	12	0	0	15	0	5	0	1	5	10	0,33	2	10	0,17	3	11	0,21
3	0	12	0	0	15	0	0	12	0	0	15	0	5	0	1	5	10	0,33	1	10	0,09	5	10	0,33
4	0	12	0	0	15	0	0	12	0	0	15	0	5	0	1	5	10	0,33	1	11	0,08	6	8	0,43
5	0	12	0	0	15	0	0	12	0	0	15	0	5	0	1	5	10	0,33	0	11	0	7	8	0,47
6	0	12	0	0	15	0	0	12	0	0	15	0	5	0	1	5	10	0,33	1	10	0,09	7	8	0,47
7	0	12	0	0	15	0	0	12	0	0	15	0	5	0	1	5	10	0,33	2	10	0,17	8	6	0,57
8	0	12	0	0	15	0	0	12	0	0	15	0	5	0	1	5	10	0,33	1	10	0,09	10	5	0,67
9	0	12	0	0	15	0	0	12	0	0	15	0	5	0	1	5	10	0,33	0	11	0	11	4	0,73
10	0	12	0	0	15	0	0	12	0	0	15	0	5	0	1	5	10	0,33	1	10	0,09	11	4	0,73

Tabulka 4 – test vlivu statické zajímavosti



Obrázek 6 – graf poměru předmětů v paměti v závislosti na čase

Vysvětlivky k tabulce a grafu

Sloupce 1-3 je test 1 pro krátkodobou paměť, 1 - počet truhel, 2 - počet mečů a 3 - poměr počtu truhel k součtu truhel a mečů, sloupce 4-6 dtto pro dlouhodobou paměť, sloupce 7-9 je test 2 pro krátkodobou paměť atd.

Křivky v grafu, které nejsou vidět, jsou nulové.

Test ukázal velkou citlivost paměti na nastavení. Velká hodnota zajímavosti v prvním testu neumožnila truhlám dostat se do paměti a meče ovládly obě paměti, protože dlouhodobá paměť neměla žádný předmět, který by mohla svou unikátností upřednostnit před meči.

Druhý test dopadl stejně s výjimkou, že posteli se podařilo dostat jednou do krátkodobé paměti. Tady se ukázalo, že ani výraznější zohlednění množství stejných předmětů nepomůže, protože v každé místnosti je přibližně stejně mečů a truhel a tím k tomu, že při odstraňování mají nižší hodnotu, protože zajímavost je vyšší.

U třetího testu je podobný průběh, pouze je ovlivněn menším počtem předmětů v lokacích, proto se mezi nultým a prvním dnem projevilo v krátkodobé paměti, která se později stala dlouhodobou, vyvažování stejných předmětů. Dále se vždy při přidávání obsahu krátkodobé paměti do dlouhodobé srovnaly hodnoty truhel a mečů a vyhrály meče pro svou zajímavost.

U čtvrtého testu nastal jiný jev. Díky nižší hodnotě zajímavosti mečů se pár truhel dostalo do krátkodobé paměti a z ní do dlouhodobé. Díky nižšímu počtu byly truhly při přesunu do dlouhodobé paměti zvýhodněny a začaly dokonce meče vytlačovat z dlouhodobé paměti. Tento trend se ke konci začal stabilizovat, protože přísun nových mečů a truhel vyvážil předměty, původně umístěné v dlouhodobé paměti a při odstraňování se rozhodovalo i podle dalších kritérií (např. umístění předmětu u vstupu do hor, kdy je spatřen při návratu a tím má nejmenší čas od posledního spatření).

6.5 Závěr testů

Výsledky testů ukázaly na potřebu dobrého nastavení paměti a při generování světa mít zajištěné i mechanismy upravující výskyt předmětu, kdy více předmětů s velkou zajímavostí je schopno narušit funkci paměti. Současně byl prokázán negativní vliv podobných velikostí krátkodobé a dlouhodobé paměti. Pokud v krátkodobé paměti jeden typ předmětů převážil a zároveň v dlouhodobé paměti byl podobný poměr předmětů, potom se po přidání vzpomínek z krátkodobé paměti příliš zvýhodnily

předměty, které byly v obou pamětech v menšině a byly schopné postupem času zvrátit poměr v dlouhodobé paměti. Tyto problémy by se daly odstranit změnou odebírání předmětů. Současné odebírání předmětů s nejnižšími hodnotami nahradit odebráním jednoho předmětu s nejnižší hodnotou s následným přepočítáním hodnoty. Postup opakovat do té doby, až se počet předmětů sníží na požadovanou úroveň. Toto řešení je špatně použitelné, protože by se zvýšila náročnost výpočtu z $O(n)$ na $O(n^2)$.

7. Závěr

V rámci této práce byl navržen model paměti a implementován do projektu IVE. Navržený model používá dvě oddělené části. Jednu pro nové předměty a druhou pro již zapamatované. Obě části se jednou za čas synchronizují, aby nové věci šly do již zapamatovaných. Pro ohodnocení vzpomínky se používá četnost spatření, čas od posledního spatření, zajímavost objektu a počet stejných předmětů v paměti.

Pro implementování paměti bylo rozšířeno prostředí IVE o sledování paměti postav, zdokonalené rozhraní pro volitelné předměty nebo snadnější zapnutí detailní simulace světa. Paměť se vyskytuje buď jako smysl nebo přímo v postavě rozšířené o referenci na paměť. Tento způsob byl zvolen, aby části kódu, které používají senzory, mohly pracovat i s paměti.

Testy ukázaly dobré chování v různorodém prostředí, naopak v místech s velkým počtem stejných předmětů bylo chování neuspokojivé.

7.1 Zkušenosti z vývoje

Zkušenosti s tímto typem vývoje byly pro mě velmi podnětné. Nejprve bylo nutné pochopit na jakých principech funguje původní kód a teprve následně se pustit do samotného vývoje. V rámci této práce jsem se poprvé setkal s programovacím jazykem *java* a ocenil jsem, že k vývoji byl použit tento jazyk, protože ladění takto rozsáhlého programu psaného v jazyce *C* nebo *C++* by bylo podstatně náročnější.

Úskalí uprav a zásahů do již existující aplikace se ukazovaly po celou dobu vývoje. Různé věci nefungovaly tak, jak jsem si zprvu myslel. Po změně některých částí programu, u kterých jsem předpokládal, že s danou věcí nesouvisí, se přestala simulace chovat způsobem, jakým jsem očekával. Nejzávažnější problém nastal v okamžiku, kdy byla práce hotova a již se prováděly pouze testy. V jednom testovacím scénáři přestala chodit postava. Nakonec se ukázalo, že je chyba v původním programu a díky spolupráci s autory původní aplikace byla odstraněna.

7.2 Možná budoucí rozšíření

Pokud by se IVE rozšířilo o možnost komunikace mezi postavami a tato komunikace byla implementována pomocí potomků IVE objektů, pak by současná paměť byla schopna si komunikaci také zapamatovat. Tím by se dalo vyzkoušet, zda je model paměti vhodný i na toto použití a zároveň jestli je paměť schopna odpovídat uspokojivě na dotazy.

Další možností je otestovat, zda by pomohlo pro více postav vytvořit společnou datovou strukturu vzpomínek a v paměti by na ni byly pouze odkazy. Tyto vzpomínky by byly neměnné a v případě aktualizace by se vytvořila nová vzpomínka v datové struktuře a nahradila by původní odkaz. Toto řešení by mohlo ušetřit místo v paměti počítače, pokud si postavy pamatují podobné věci, které se moc nemění. Stejně zapamatované věci by byly uloženy jednou a pouze by na ně bylo více odkazů.

Doplnit do projektu IVE prohledávání neznámých prostor popsaných v kapitole 5.2 *Hledání nových předmětů*.

8. Literatura

- [1] Badelley Alan: Vaše Pamět, Jota, Brno, 1996
- [2] AKRI Ltd. Applied Knowledge Research & Innovation:
Cognition – Memory - Human Memory Model,
<http://www.akri.org/cognition/hummod.htm>
- [3] Trapp Dave: Theory of memory,
<http://homepage.mac.com/dtrapp/essays/memory.html>
- [4] Smolík P.: Spánek a sny – neurofyziologie, neuroanatomie, neurochemie.
cirkadiánní rytmicita, evoluce. Psychiatrie 2002, strany 121–129.
- [5] Kubr, Kulhánek, Poch, Šafrata, Šerý, Šulc: Dokumentace k projektu IVE
- [6] Sutton John: Standford encyclopedia of phylosophy – Memory,
<http://plato.stanford.edu/entries/memory/>
- [7] Rakesh Mohan Hallen: Bewildering brain,
<http://www.scribd.com/doc/27937/Bewildering-Brain/>
- [8] Bojar, Brom, Hladík, Toman, Vejlupek, Voňka : Enti – teoretická dokumentace,
<http://ufal.mff.cuni.cz/~bojar/enti/publications/teorie.pdf>
- [9] Pešková Klára: Specifikace projektu potvůrka,
<http://carolina.mff.cuni.cz/~peskova/bakalarka/index.php?p=spec>

9. Seznam příloh

Přílohou bakalářské práce je CD s aplikací, testovacími scénáři, uživatelskou a programátorskou dokumentací a soubory s výsledky testů.