

HERBAL

Projektová dokumentácia

Vedúci projektu	Ing. Pavel Bartoš
Členovia tímu č. 19	Bc. Miloš Auder
	Bc. Andrej Belica
	Bc. Lukáš Ďurčák
	Bc. Miroslav Mikuláš
	Bc. Martin Paššák
	Bc. Ján Romaňák

Obsah

0	Úvod	0-1
0.1	Účel dokumentu.....	0-1
0.2	Slovník používaných pojmov.....	0-1
1	Analýza	1-1
1.1	Hypotézy	1-3
2	Špecifikácia požiadaviek.....	2-1
2.1	Funkcionálne požiadavky	2-1
2.1.1	Svet	2-1
2.1.2	Herb	2-1
2.1.3	Gén.....	2-2
2.1.4	Potrava.....	2-4
2.1.5	Požiadavky na program a vizualizáciu	2-4
2.2	Nefunkcionálne požiadavky	2-5
2.3	Do budúca	2-5
3	Návrh.....	3-1
3.1	Návrh jadra systému	3-1
3.1.1	Model prípadov použitia.....	3-1
3.1.2	Model tried	3-4
3.2	Návrh vizualizácie.....	3-6
3.2.1	Model tried	3-6
3.2.2	GUI	3-7
3.2.3	Vizualizácia sveta	3-7
	Dokumentácia riadenia projektu	1
4	Úvod k riadeniu projektu.....	4-1
5	Ponuka.....	5-1
6	Projektový plán	6-1

7	Úlohy členov v tíme.....	7-1
8	Záznamy zo stretnutí.....	8-1
9	Metodiky	9-1
9.1	Metodika pre písanie zdrojového kódu	9-1

0 Úvod

0.1 Účel dokumentu

Tento dokument pozostáva z dvoch častí. Prvá časť obsahuje analýzu, špecifikáciu, návrh a opis implementácie programu vyvíjaného v rámci projektu HERBAL. Druhá časť predstavuje dokumentáciu riadenia tímu počas projektu. Projekt je vypracovaný v rámci predmetu Tímový projekt na Fakulte Informatiky a Informačných Technológií Slovenskej Technickej Univerzity v Bratislave. Rolu zákazníka/vedúceho projektu zastáva Ing. Pavel Bartos. Tím pozostáva z vedúceho tímu Bc. Martina Paššáka a ďalších členov tímu Bc. Miloša Audera, Bc. Andreja Belicu, Bc. Lukáša Ďurčáka, Bc. Miroslava Mikuláša a Bc. Jána Romaňáka.

0.2 Slovník používaných pojmov

Herb:

Pojem herb označuje akýkoľvek organizmus v programe, ktorý obsahuje genetický kód a v každom kroku simulácie vykonáva na základe tohto kódu isté činnosti.

Potrava:

Pojem potrava označuje základný zdroj energie pre herbov. Potrava môže ale nemusí obsahovať vlastný genetický kód a vykonávať akcie počas behu simulácie.

1 Analýza

Umelá inteligencia sa ako vedný odbor objavila v polovici 50.-tych rokov minulého storočia – samotný pojem sa prvýkrát objavil na konferencii v Dartmouth College v júli 1956 [1]. Odvtedy boli hlavnými záujmami umelej inteligencie skúmanie a návrh inteligentných agentov – agentov, ktorý by boli schopní vnímať svoje okolie a vykonávať rôzne kroky za účelom toho, aby maximalizovali efektívnosť svojej práce. V oblasti sa vytvorili dva prístupy k vytvoreniu inteligencie – prístup zhora-nadol a prístup zdola-nahor. Samotná oblasť sa takisto rozdelila na veľké množstvo „smerov“. Z ich postupným rozvojom sa vedci pokúšali o čoraz komplexnejšie simulácie a po čase narazili na hardvérovú prekážku v podobe fyzických robotov reprezentujúcich agenty. Aby sa táto prekážka obišla, začalo sa pristupovať k simuláciám agentov v počítačových systémoch. Tak sa v 80.-tych rokoch objavila aj oblasť umelého života.

Umelý život (ang.: Artificial Life, Alife) je „oblasť výskumu zameraná na porozumenie života prostredníctvom pokusov o abstrakciu základných dynamických princípov za týmto biologickým fenoménom a na rekonštrukciu tohto dynamického systému v inom médiu tak, aby ho bolo možné študovať a manipulovať s ním novými spôsobmi“[2]. Táto oblasť bola takto pomenovaná Christopherom Langtonom v roku 1986. Umelý život sa delí na tri podskupiny – soft Alife (založený na softvérovej reprezentácii), hard Alife (hardvérová reprezentácia) a wet Alife (biochemicky vytvorený umelecký život). Pojem umelý život sa často používa aj na označenie samostatnej prvej skupiny – softvérovo reprezentovaného umelého života. Práve tento druh je predmetom tohto projektu. Umelý život sa od mnohých oblastí umelej inteligencie odlišuje tým, že používa prístup zdola-nahor – inteligentné správanie sa snaží dosiahnuť prostredníctvom spolupráce jednoduchých základných zložiek.

V softvérovom ponímaní je možné umelý život naprogramovať ako sekvenčný evolučný algoritmus využívajúci istú sadu inštrukcií. V tomto projekte sa však budeme zaoberať alternatívnou reprezentáciou – pravidlovým systémom. Gény a teda aj samotné správanie organizmov bude charakterizované systémom pravidiel v tvare „*Podmienka* → *Akcia*“. Takýto pravidlový systém bude pozostávať z množiny pravidiel, ktoré sa budú skladať zo základných podmienok a akcií. Tieto podmienky a akcie budú do pravidiel komponované s úmyslom vytvoriť jedinečné vzory správania pre konkrétne implementované druhy organizmov a ich kompozícia bude teda špecifikovať jednotlivé druhy. Množiny základných podmienok a akcií budú vytvorené na základe špecifikácie požiadaviek pre projekt.

Pravidlový systém vo všeobecnosti okrem množiny pravidiel vyžaduje aj existenciu inferenčného mechanizmu, ktorý rozhoduje o tom, ktoré konkrétne pravidlá z množiny znalostí sa využijú v jednotlivých krokoch spracovania vstupov. Existujú dva základné druhy inferenčných mechanizmov – sekvenčný a náhodný. Sekvenčný mechanizmus prechádza pravidlá sekvenčne a na aplikáciu vyberie prvé pravidlo, ktoré spĺňa podmienku. Náhodný mechanizmus narozdiel od sekvenčného prechádza pravidlá v náhodnom poradí. Kombináciou týchto dvoch prístupov je možné vytvoriť ďalšie druhy mechanizmov.

V rámci tohto projektu bude nutné vytvoriť model sveta, v ktorom sa bude umelý život rozvíjať. Tento model je možné charakterizovať ako systém, v ktorom bude interagovať veľké množstvo rôznych entít (jednotlivé organizmy, svet, potrava, zásahy od používateľa). Preto je vhodné tento model reprezentovať objektovo orientovaným prístupom. Pre úspešnú implementáciu bude potrebné vytvoriť nasledovné druhy tried:

- organizmus – skupina/hierarchia tried, ktoré budú reprezentovať jednotlivé organizmy a druhy organizmov; tieto triedy budú charakterizovať ich správanie (gény)
- svet – tieto triedy budú reprezentovať svet v podobe mriežky, budú umožňovať existenciu a pohyb organizmov v priestore
- používateľské rozhranie – triedy umožňujúce používateľovi monitorovať priebeh simulácií umelého života a prípadne do nich robiť zásahy, zmeny parametrov
 - vizualizácia – dôležitá súčasť rozhrania, ktorá umožní prehľadné sledovanie simulácií a prípadnú interakciu používateľa so svetom
- kontrolný mechanizmus – tieto triedy budú zodpovedné za riadenie sveta a organizmov – t.j.:
 - inicializáciu a spravovanie populácií/sveta
 - aplikovanie mechanizmu času na organizmy/svet
 - riadenie poradia, v ktorom budú organizmy vykonávať svoje aktivity (prípadne zabezpečenie paralelizmu pre tento problém)
 - riadenie inferenčného mechanizmu pravidlového systému génov organizmov
 - riadenie rozhrania a vizualizácie

- logovanie – triedy zabezpečujúce zaznamenávanie priebehu simulácií; tieto informácie umožnia vykonávanie analýzy dát na základe zvolených zachytených štatistických údajov

Vymenované skupiny tried bude vhodné implementovať v rámci samostatných modulov, ktoré budú len minimálne previazané. Takéto riešenie je vhodné, lebo umožňuje jednoduché pridávanie alebo zmenu individuálnych komponentov, ktoré neovplyvní činnosť celého systému. S týmto riešením je takisto spojená nutnosť vytvoriť kvalitné rozhrania medzi modulmi tak, aby bolo možné ľahko prenášať medzi modulmi potrebné dáta.

Pri implementácií bude nutné klásť dôraz aj na efektivitu vykonávania aplikácie. Hlavným jej účelom totiž bude vykonávanie simulácie, z ktorých sa budú vyvodzovať rôzne závery. Aby bolo možné vyvodiť tieto závery kvalitne, bude potrebné pracovať s čo možno najväčšími populáciami a veľkosťou sveta a tieto môžu mať značný negatívny vplyv na rýchlosť vykonávania programu. Náhodný charakter simulácií takisto vyžaduje ich mnohonásobné vykonávanie aby bolo možné získané dáta považovať za dostatočne reprezentatívne – toto je ďalší dôvod zväčšujúci dôležitosť rýchlosti vykonávania simulácií.

1.1 Hypotézy

Ako už bolo vyššie naznačené, jedným z hlavných výstupov aplikácie budú zaznamenané štatistické údaje zo simulácií, ktoré umožnia štúdium vplyvu rôznych faktorov (prítomnosť/neprítomnosť istých génov, počet génov, ...) na úspešnosť populácií. Aby bolo možné výsledky študovať a vyvodiť z nich závery bude nutné vykonať veľké množstvo simulácií, avšak my sme sa rozhodli aj bez informácií z týchto simulácií vytvoriť vlastné hypotézy o tom, ktoré faktory a akým vplyvom ovplyvnia populácie najviac. Tu uvádzame naše hypotézy; tieto hypotézy vychádzajú z predpokladu, že podmienky simulácie sú totožné so základnými podmienkami uvedenými v základnej verzii špecifikácie požiadaviek v kapitole 2. Keďže táto špecifikácia je relatívne jednoduchá, je len ťažko možné vytvoriť veľmi odlišné hypotézy. Preto sa v našich hypotézach skôr sústreďujeme na to, ktorý faktor/faktory budú mať najzásadnejší vplyv na istý ukazovateľ populácie (napríklad priemerný vek alebo veľkosť populácie).

Janči – Economical Explorer:

V mojej hypotéze sa sústredím na to, ktoré faktory umožnia populácií dosiahnuť čo najvyšší možný priemerný vek. Predpokladám, že populácie dožívajúce sa priemerne vyššieho veku sa budú na rozdiel od ostatných populácií vyznačovať týmito znakmi:

- vyrovnaný počet génov pre otočenie doprava a doľava; zabráni to krúženiu jedinca v malej oblasti a umožní mu nájsť viac potravy
- malá priemerná cena akcií v jednotkách energie v rámci DNA, t.j.: malý počet génov obsahujúcich energeticky náročné operácie ako napríklad CLONE, CROSS, ...
- v prípade výskytu podmienky SEE_FRIEND bude akcia otočenie (do ľubovoľnej strany) – ak je pred jedincom iný jedinec, je dobrá šanca na to, že tam už nebude potrava, preto sa treba snažiť zmeniť smer pohybu jedinca

Miloš – Energy Saver:

Predpokladám, že herbovia budú úspešnejší ak budú dosahovať väčší priemerný vek. To sa však nemôže vťahovať na celú populáciu.

- novým jedincom bude postupne miznúť podmienka NILL a pribúdať akcia WAIT
- druh s väčšou začiatočnou rozmanitosťou akcií bude úspešnejší

Miki – Fat Herbs:

Predpokladám, že keď herbovia dostanú priveľa potravy postupom času zlenivejú a takmer stratia schopnosť pohybovať sa.

- Čím viac jedincov v populácii, tým nižší priemerný vek, naopak, čím menej jedincov v populácii, tým vyšší priemerný vek
- Čím bude viac potravy, tým menej sa bude vyskytovať akcia pohybu a naopak čím menej potravy, tým viac akcií pohybu a otáčania

Lukáš – Survival Instinct:

Predpokladám, že pri dosiahnutí ustáleného stavu organizmov bude možné určiť stratégiu prežitia

- vzniknuté organizmy budú buď aktívne (snažia sa prejsť čo najväčšiu časť mapy) alebo pasívne (šetria energiu – hýbu sa len v nutných prípadoch)
- schopnosť týchto organizmov prežiť bude približne rovnaká

Maťo – Vision Specialization:

Moja teória sa zaoberá senzormi a týka sa sveta, v ktorom budú zberači aj lovci.

- Zberači budú mať väčšiu schopnosť prežiť ak budú vidieť všetky políčka okolo seba (čo najväčší zorný uhol)
- Lovci budú mať väčšiu schopnosť prežiť ak budú vidieť viac políčok jedným smerom (ak bude možné viac ako jedno pred seba, inak im bude stačiť políčko pred sebou a 2 diagonálne vedľa nich, ostatne už nebudú mať vplyv na úspešnosť jedinca)

Zdroje:

[1]Brunette, E.S.; Flemmer, R.C.; Flemmer, C.L.: A review of artificial intelligence, *4th International Conference on Autonomous Robots and Agents 2009*, pp.385-392, 2009

[2]Langton, C. G.: Preface In *Artificial Life II, SFI Studies in the Sciences of Complexity*, vol. 10, Addison-Wesley, 1992.

2 Špecifikácia požiadaviek

Vychádzajúc zo zadania, pracovného textu ktorý sme obdržali a z diskusií, sme identifikovali tieto požiadavky:

2.1 Funkcionálne požiadavky

V ďalšom texte budú bližšie popísané požiadavky na jednotlivé časti vyvíjaného systému, ktorými sú svet, herb, potrava a vizualizácia.

2.1.1 Svet

Svet HERBAL-u je reprezentovaný ako dvojrozmerná obdĺžniková plocha, rozdelená mriežkou s rozmermi $m \times n$. Každé políčko je identifikované svojou x -ovou a y -ovou súradnicou. Priestor sveta je súvislý, t.j. neexistujú v ňom izolované oblasti. Na jednom políčku mriežky sa môže nachádzať výhradne iba jeden objekt nasledujúcich typov:

- Herb
- Potrava
- Stena (predstavuje zároveň koniec sveta)
- Prázdne políčko

V tomto svete môžu herbovia vykonávať všetky akcie, ktoré majú zapísané v chromozóme. Tieto akcie sú bližšie špecifikované nižšie (napr. pohyb, otáčanie, kríženie...).

Plynutie času vo svete sa deje po tikoch (základná časová jednotka). Určitý počet tikov sa dá vyjadriť ako jeden deň, a určitý počet dní ako jeden rok.

2.1.2 Herb

Každý herb je jedinečný organizmus a jeho stav je reprezentovaný týmito vlastnosťami:

- DNA (chromozóm). Ide o štruktúru v ktorej sú za sebou uložené jednotlivé gény. Každý herb má po celý život nemennú DNA. Počet génov v chromozóme je jedným z parametrov sveta. Pre herbov jedného druhu je vždy rovnaký.
- Vek. Vyjadruje dĺžku života jedinca v dňoch.

- Energia. Vyjadruje vitalitu herba, ktorá sa časom mení. Energiu môže dopĺňať konzumovaním potravy. Energia herba nemôže prekročiť maximálnu hranicu, ktorá sa stanovuje ako parameter. Naopak, energiu stráca vykonávaním akcií, ale aj každým uplynutým dňom (konštantnú hodnotu).
- ID. Jedinečná identifikácia herba v populácii.
- Pozícia. Udáva momentálne umiestnenia herba vo svete. V 2D svete sú to dve súradnice x a y.
- Orientácia. Každý herb je otočený v určitom smere. Vo svete HERBAL to môžu byť štyri rôzne smery – sever, juh, východ, západ (resp. hore, dole, vpravo, vľavo)

2.1.3 Gén

Gén je štruktúra, ktorá pozostáva z ľavej a pravej strany. Na ľavej strane sa nachádzajú podmienky, ktoré môžeme chápať ako senzory jedinca. Na pravej strane sa nachádzajú akcie, ktoré môže jedinec vykonávať. Pri výbere génov je dôležité ako sú za sebou v chromozóme usporiadané.

Ľavá strana pravidla

Na ľavej strane pravidla herba sa nachádzajú podmienky (senzory). Jeden druh má k dispozícii rovnakú sadu testov. Táto sada je jedným z parametrov sveta. Na ľavej strane pravidla sa nachádza 1 až m (parameter) podmienok.

Existuje základná sada podmienok:

- NIL. Predstavuje prázdnu podmienku – žiaden senzor. Vždy sa automaticky vyhodnocuje ako TRUE. Ak sa nachádza pri iných podmienkach, tak sa vyhodnocovanie ľavej strany génu prakticky o jednu podmienku skraca.
- FALSE. Vždy sa vyhodnocuje ako nesplnená, teda natrvalo blokuje tento gén.
- SEE_FOOD. Vracia TRUE, ak sa priamo pred herbom nachádza políčko na ktorom je potrava.
- SEE_END_OF_WORLD. Reprezentuje senzor, ktorý vie rozoznať či je bezprostredne pred herbom políčko, kde sa nachádza prekážka.
- SEE_FRIEND. Táto podmienka sa vyhodnotí ako pravdivá, ak sa pred herbom nachádza iný herb, ale rovnakého druhu.

- SEE_HERB. Táto podmienka sa vyhodnotí ako pravdivá, ak sa pred herbom nachádza herb hocikákeho druhu.
- SEE_EMPTY. Senzor zisťujúci, či je pred herbom prázdne políčko.

Pravá strana pravidiel

Na pravej strane pravidiel herba sa nachádzajú akcie, teda činnosti ktoré je herb schopný vykonať. Každý herb rovnakého druhu disponuje rovnakou sadou akcií, pričom táto sada je tiež jedným z parametrov sveta. Na pravej strane pravidiel sa môže nachádzať 1 až n (parameter) akcií.

Vo svete HERBAL sa vychádza z nasledovnej minimálnej sady akcií, pričom vykonanie každej akcie trvá určený počet tikov a na jej vykonanie sa spotrebuje určené množstvo energie:

- NULL. Ide o prázdnu akciu, teda akciu, ktorá nič nevykoná. Nespotrebuje sa pri nej žiadna energia a neminie sa žiadny čas, pretože táto akcia je preskočená.
- WAIT. Reprezentuje akciu čakania, táto akcia sa vykoná, ale herb nerobí nič, len čaká. Na túto akciu herb spotrebuje minimum energie a táto akcia trvá zvyčajne jeden deň, môže však aj menej.
- TURN_LEFT. Herb sa na mieste otočí o 90° vľavo, teda proti smeru hodinových ručičiek. Po vykonaní tejto akcie herb ostáva na rovnakom políčku ako pred jej vykonaním. Akcia by mala byť vykonateľná kedykoľvek a spotrebuje sa na ňu malé množstvo energie a trvá spravidla len jeden tik.
- TURN_RIGHT. Herb sa na mieste otočí o 90° vpravo, teda v smere hodinových ručičiek. Po vykonaní tejto akcie herb ostáva na rovnakom políčku ako pred jej vykonaním. Akcia by mala byť vykonateľná kedykoľvek a spotrebuje sa na ňu malé množstvo energie a trvá spravidla len jeden tik.
- MOVE. Herb sa pohne vpred o jedno políčko v tom smere, v ktorom je práve otočený. Na túto akciu herb spotrebuje časť energie a minie časť času bez ohľadu na to, či sa skončila úspechom. Túto akciu je možné vykonať len v prípade, keď sa pred herbom nachádza prázdne políčko.
- CLONE. Pri vykonaní tejto akcie sa herb rozdelí na dvoch identických jedincov, pričom sa spotrebuje všetok zvyšný čas herba a veľké množstvo energie. Ak chce herb vykonať túto akciu, musí mať aspoň stanovené množstvo energie a musí mať vedľa seba aspoň jedno

políčko voľné. Novovzniknutý herb sa narodí s identickou dna, ktorá ale ešte v rámci jeho zrodu zmutuje a tým zmení svoju štruktúru. Nový herb sa zrodí na náhodnom voľnom políčku vedľa pôvodného herba.

- CROSS. Predstavuje kríženie dvoch herbov, ktoré ak je úspešné, tak vznikne nový herb. Pri tejto akcii sa spotrebuje všetok zvyšný čas v rámci dňa a veľké množstvo energie. Aby bolo možné vykonať kríženie, musí sa vedľa herba nachádzať iný herb rovnakého druhu, musí byť vedľa herba voľné aspoň jedno políčko a obidvaja herbovia musia mať istú minimálnu energiu a vek. Po úspešne vykonanom krížení sa na náhodnom políčku vedľa herba narodí nový jedinec, ktorý získa od rodičov jeho počiatočnú energiu a skrížený chromozóm, ktorý môže byť ešte aj zmutovaný.

2.1.4 Potrava

Potrava vo svete HERBAL predstavuje zdroj energie pre herbov, teda niečo, bez čoho herbovia nemôžu prežiť a sú nútení potravu hľadať a konzumovať. Vo svete existuje len jeden druh potravy, ktorý má konštantné množstvo energie, ktoré sa môže herbovi doplniť po jej skonzumovaní. Potrava sa zrodí vždy iba raz za isté časové obdobie, ktoré je jedným z parametrov. Počas tohto obdobia sa naraz objavia všetky plody, pričom sa môžu objaviť len na prázdnom políčku. Ak počas sezóny zostane nejaká potrava neskonzumovaná, táto potrava sa neskazí, ale zostáva pre herbov k dispozícii. V okamihu, keď herb prejde po políčku kde je potrava, tak ju celú skonzumuje, bez ohľadu na to, koľko energie skutočne potrebuje.

Počet plodov, ktoré sa objavia pri dozrievaní, je tiež jedným z parametrov sveta. Tento počet taktiež závisí od počiatočného počtu plodov a od zvolenej stratégie dozrievania.

2.1.5 Požiadavky na program a vizualizáciu

- Vizualizácia sa musí dať hocikedy vypnúť a znova zapnúť a to z dôvodu, že vizualizácia beh programu do značnej miery spomaľuje a nie vždy je pre používateľa potrebná.
- Počas behu by sa mali dať sledovať aktuálne informácie o svete.
- Každý herb by mal byť označiteľný tak, aby sa o ňom dali počas simulácie sledovať detailné informácie.
- Interaktívne zasahovanie do sveta (vloženie nášho vlastného herba – hrdinu, prírodná katastrofa a pod.)

- Možnosť sledovať rôzne štatistiky.
- Možnosť uložiť si nastavenia sveta a neskôr ich načítať.
- Sledovanie veľkého sveta – priblíženie (ZOOM).
- Ukladanie dát o svete a štatistik do súboru.

2.2 Nefunkcionálne požiadavky

System je navrhnutý ako desktopová aplikácia. Používateľské rozhranie musí byť vysoko intuitívne, aj napriek tomu, že by malo umožňovať nastavovať veľké množstvo parametrov. Keďže dôležitou požiadavkou špeciálne pre náš projekt je rýchlosť, je potrebné vhodne zvoliť jazyk a vývojové prostredie. Našou požiadavkou nie je vytvorenie multiplatformovej aplikácie, preto sme sa rozhodli pre C# a vývojové prostredie Microsoft Visual Studio 2008. Vizualizácia je dôležitou súčasťou aplikácie, mala by byť pre používateľa príjemná na pohľad a ľahko manipulovateľná. Toto nám umožňuje framework XNA, ktorý výborne spolupracuje práve s jazykom C#, vzhľadom na to že ide tiež o produkt z dielne Microsoftu.

2.3 Do budúca

- Plody potravy s rozdielne veľkou energiou.
- Zjedenie potravy ako explicitná akcia.
- Potrava bude mať sadu génov, pomocou ktorých by sa potrava rozmnožovala, teda rozširovala by sa do susedných políčok.
- Herb by nezjedol celý plod, ale iba časť z neho a istá zostávajúca časť plodu by zostala k dispozícii iným herbom.
- Podmienka zisťujúca či má herb stanovenú úroveň energie.
- Podmienka testujúca aktuálny vek herba.
- Herb bude vidieť aj šikmo.
- Herb bude vidieť na ďalej ako jedno políčko.
- Odovzdanie potravy (energie) inému herbovi rovnakého druhu v bezprostrednej blízkosti.

- Položenie potravy na voľné políčko.
- Modifikovanie sveta.
- Explicitná akcia – smrť.
- Pridanie zvukov a efektov.
- Iný tvar políčok ako štvorec.

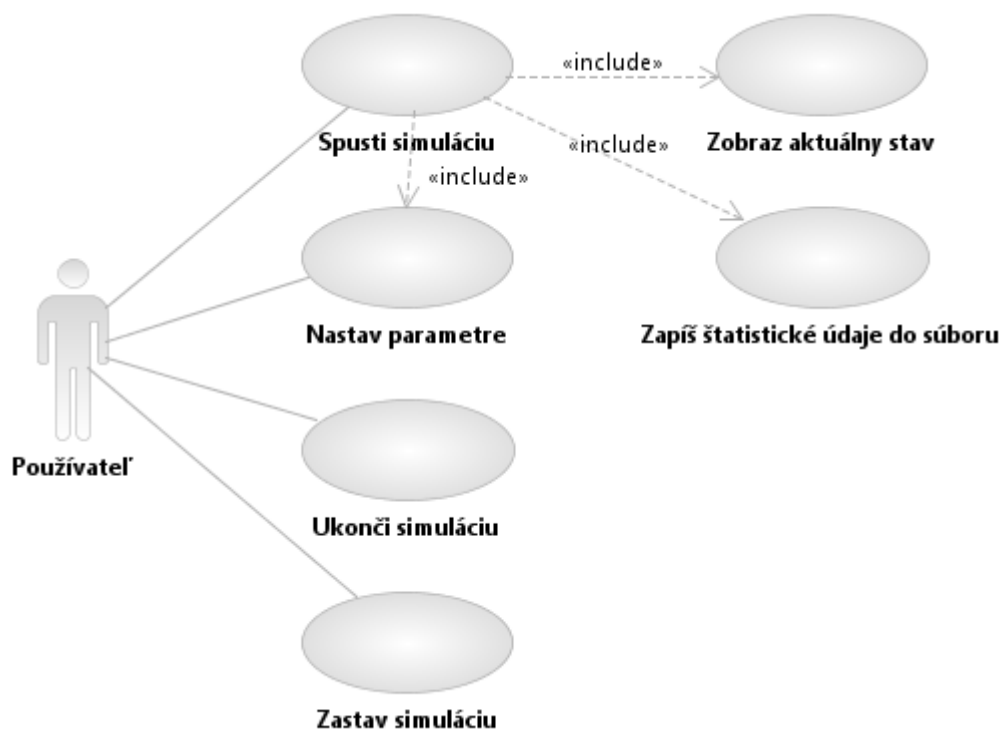
3 Návrh

3.1 Návrh jadra systému

V tejto časti návrhu opisujeme časť modelu systému, ktorá je zodpovedná za aplikačnú logiku a riadenie programu. Túto časť návrhu vypracovali Andrej Belica, Martin Paššák a Ján Romaňák.

3.1.1 Model prípadov použitia

Na začiatku uvádzame model prípadov použitia. Diagram je znázornený na obrázku (Obr. 1).



Obr. 1 – Diagram prípadov použitia

Spusti simuláciu:

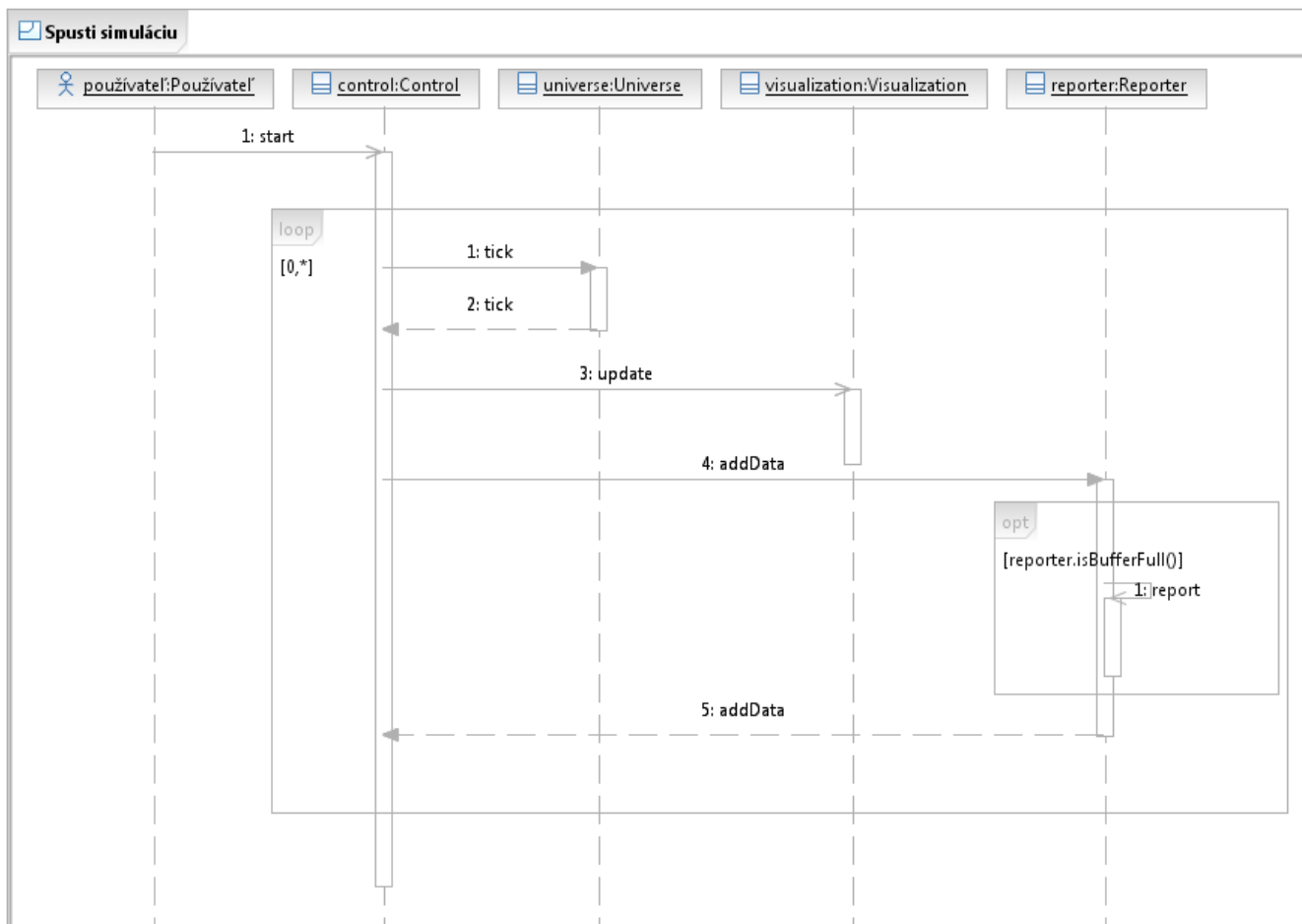
V tomto prípade použitia vystupuje používateľ a po uskutočnení prípadu použitia bude v systéme spustená simulácia.

Používateľovi sa zobrazia aktuálne parametre simulácie. Používateľ ich môže upraviť, ak to považuje za potrebné. Používateľ potvrdí parametre a potom zvolí možnosť inicializácie sveta. Následne má možnosť parametre opäť pozmeniť a svet nanovo inicializovať, alebo môže spustiť simuláciu.

Simuláciu je možné spustiť aj ak už bola v predchádzajúcom priebehu programu inicializovaná, spustená a následne pozastavená.

Keď je simulácia v stave spustená, systém v slučke vyvoláva kroky sveta, v ktorých organizmy uskutočňujú svoje akcie. Systém v týchto krokoch používa prípady použitia „Zobraz aktuálny stav“ a „Zapíš štatistické údaje do súboru“.

Priebeh akcií v rámci tohto prípadu použitia je ilustrovaný na obrázkoch (Obr. 2 a Obr. 3).

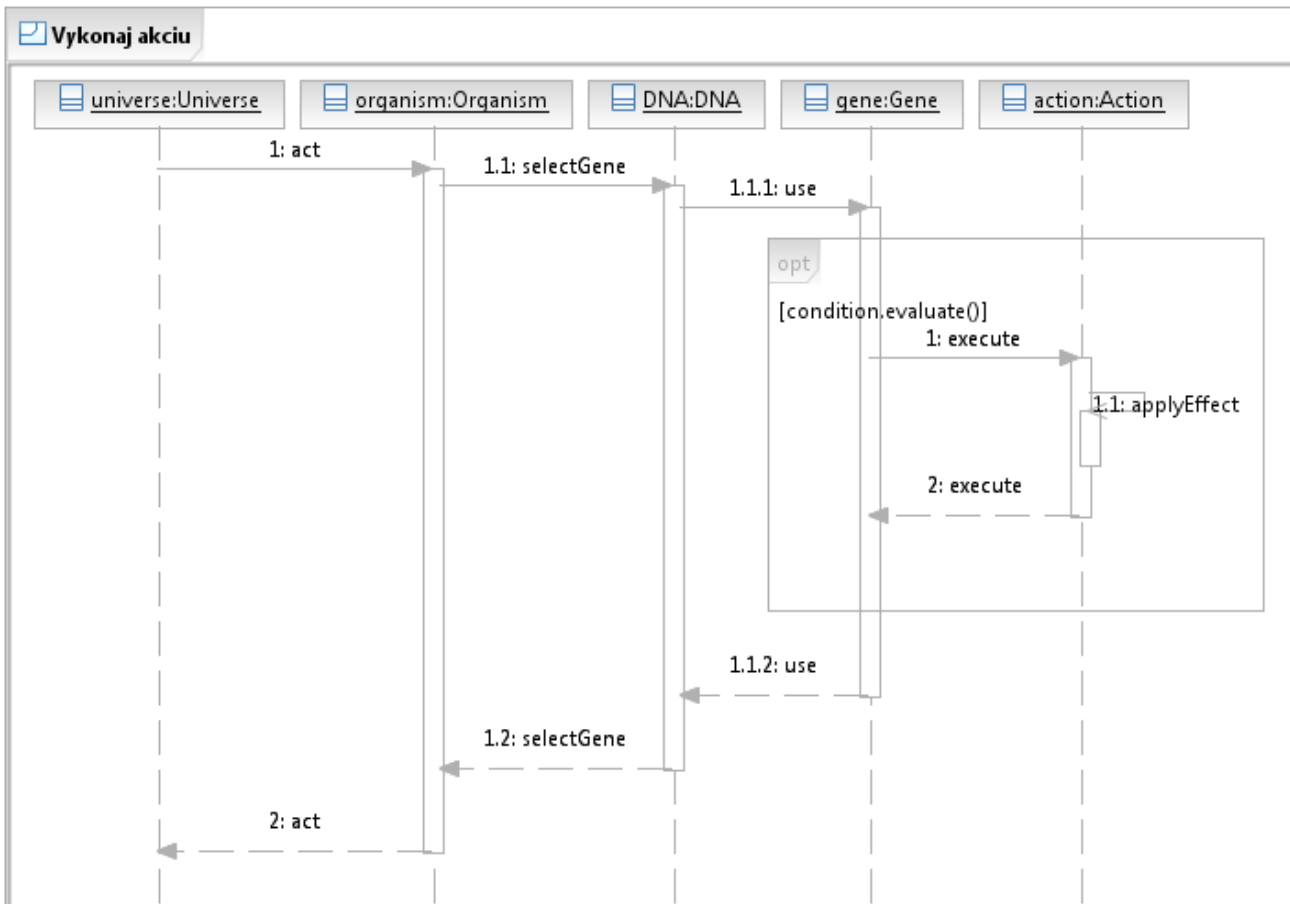


Obr. 2 – Sekvenčný diagram spustenia simulácie

Zobraz aktuálny stav:

Tento prípad použitia je zahrnutý v prípade použitia „Spusti simuláciu“.

Keď je simulácia v stave spustená, systém v každom kroku vyvoláva operáciu obnovenia vizuálnej reprezentácie sveta.



Obr. 3 – Sekvenčný diagram vykonania akcie

Zapíš štatistické informácie do súboru:

Tento prípad použitia je zahrnutý v prípade použitia „Spusti simuláciu“.

Keď je simulácia v stave spustená, systém po každom kroku odosiela aktuálny stav sveta do podsystému zodpovedného za zapisovanie štatistických informácií do súboru.

Zastav simuláciu:

V tomto prípade použitia vystupuje používateľ a po uskutočnení prípadu použitia bude simulácia v stave pozastavená.

Tento prípad použitia má za predpoklad to, že simulácia je v stave spustená. Keď používateľ zvolí možnosť zastavenia simulácie, systém dokončí momentálne vykonávaný krok a následne simuláciu pozastaví.

Ukonči simuláciu:

V tomto prípade použitia vystupuje používateľ a po uskutočnení prípadu použitia bude simulácia v stave ukončená.

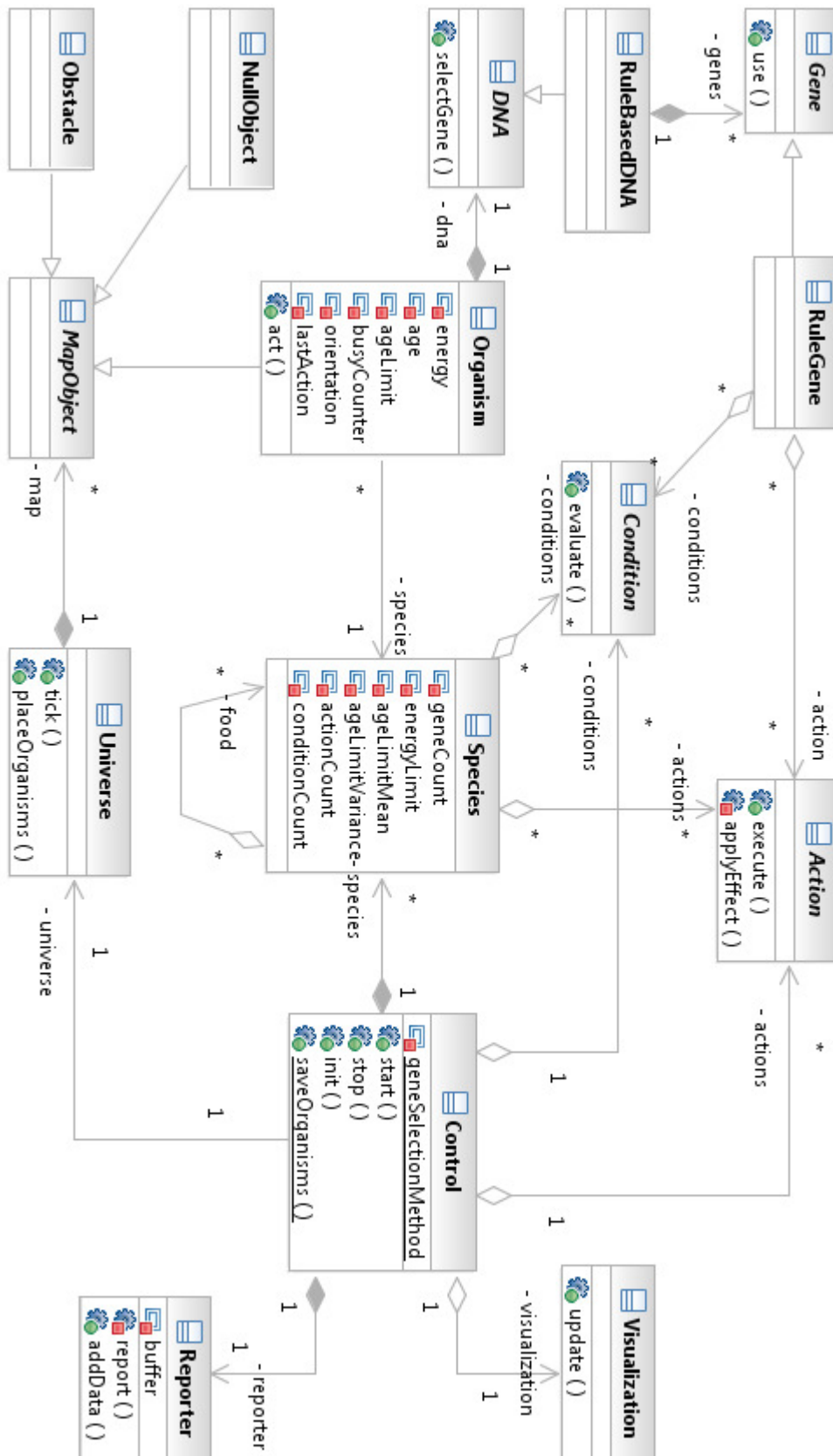
Tento prípad použitia má za predpoklad to, že simulácia je v stave spustená alebo pozastavená. Keď používateľ zvolí možnosť ukončenia simulácie, systém dokončí momentálne vykonávaný krok a následne simuláciu ukončí.

3.1.2 Model tried

V tejto podkapitole sa nachádza nami navrhovaná dekompozícia systému na triedy. Pri návrhu tejto dekompozície sme sa sústredili na to, aby bol systém ľahko rozšíriteľný. Toto sme dosiahli vytvorením abstraktných tried:

- DNA – táto abstraktná trieda umožňuje používanie a rozšírenie systému o rôzne druhy DNA organizmov. V tomto momente je jedinou konkretizáciou tejto abstraktnej triedy trieda RuleBasedDNA.
- Gene – táto abstraktná trieda umožňuje implementáciu rôznych druhov génov – napríklad základných pravidlových génov RuleGene alebo napríklad génov vyjadrujúcich rôzne atribúty organizmu (ktoré zatiaľ v návrhu neuvažujeme).
- Condition a Action – tieto abstraktné triedy umožňujú systém rozširovať o dodatočné druhy podmienok a akcií v génoch.
- MapObject – táto trieda umožňuje jednoduchú prácu so svetom obsahujúcim rôzne druhy objektov ako sú organizmy, prekážky alebo prázdne polia.

Diagram tried je uvedený na obrázku (Obr. 4).



Obr. 4 – Diagram tried jadra systému

3.2 Návrh vizualizácie

V tejto časti návrhu opisujeme vizualizáciu systému a grafické používateľské rozhranie. Túto časť vypracovali Miroslav Mikuláš, Miloš Auder a Lukáš Ďurčák.

3.2.1 Model tried

Vizualizačná časť má na starosti zobrazovanie sveta v reálnom čase. Samotná vizualizácia je volaná z hlavnej časti programu, pričom je možné nastaviť v akých časových intervaloch bude zaznamenávať zmeny sveta.

Vizualizácia je samostatná časť programu, pričom informácie, ktoré potrebuje, čerpá z hlavnej časti programu. Pre bezproblémový chod vizualizácie sú potrebné aspoň nasledujúce údaje:

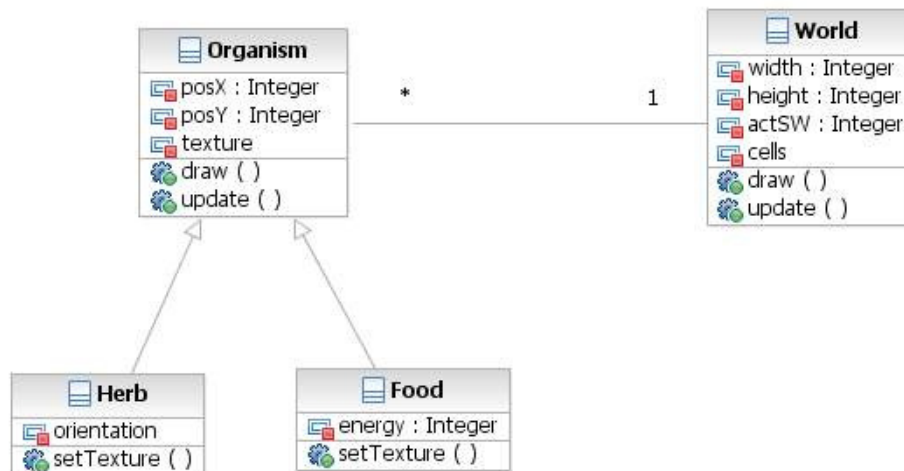
- rozmery sveta
- pozície stien
- pozície všetkých herbov
- pozície všetkých potrav
- orientácia všetkých herbov

Vďaka týmto údajom je možné vizualizovať stav sveta v každom jednom časovom okamihu. Samotný organizmus je reprezentovaný triedou Organism. Z tejto triedy sú odvodené triedy Herb a Food, pričom trieda Herb reprezentuje herba a trieda Food reprezentuje potravu. Každý organizmus má jedinečnú pozíciu vo svete, ktorá je vyjadrená atribútmi posX a posY. Aby boli organizmy na pohľad rozoznateľné, to znamená aby používateľ odlišil herba od potravu, je potrebné aby mal organizmus svoju textúru. Organizmus sa vykreslí na obrazovku procedúrou draw(). Procedúra update () sa stará o zmenu polohy organizmu, zmeny textúry, prípadne aj zničenie organizmu. Trieda, ktorá reprezentuje herba má navyše pridanú orientáciu, do ktorého smeru je herb otočený. Na základe tejto vlastnosti sa vyberie tá správna textúra. Trieda, ktorá reprezentuje potravu, má navyše vlastnosť energy. Vďaka tejto vlastnosti sa potrave zmení textúra, vďaka čomu je používateľovi na pohľad jasné, či je potrava celá, alebo už len nejaká jej časť.

Samotný svet je reprezentovaný triedou World. Svet má pevne dané rozmery, ktoré vyjadrujú vlastnosti width a height (počet políčok = width x height). Ďalším atribútom, ktorý je potrebný pre vykreslenie sveta na obrazovku, je aktuálna šírka políčka. Vďaka zmene tohto atribútu sa dá

zabezpečiť približovanie a oddiaľovanie sveta. Posledným atribútom je dvojrozmerné pole, v ktorom sú zachytené informácie o každom políčku, konkrétne či sa na ňom nachádza stena, alebo nie. Podľa tohto atribútu sa pre každé políčko správne zvolí textúra.

Jednoduchý diagram tried pre vizualizačnú časť je možné vidieť na obrázku (Obr. 5).



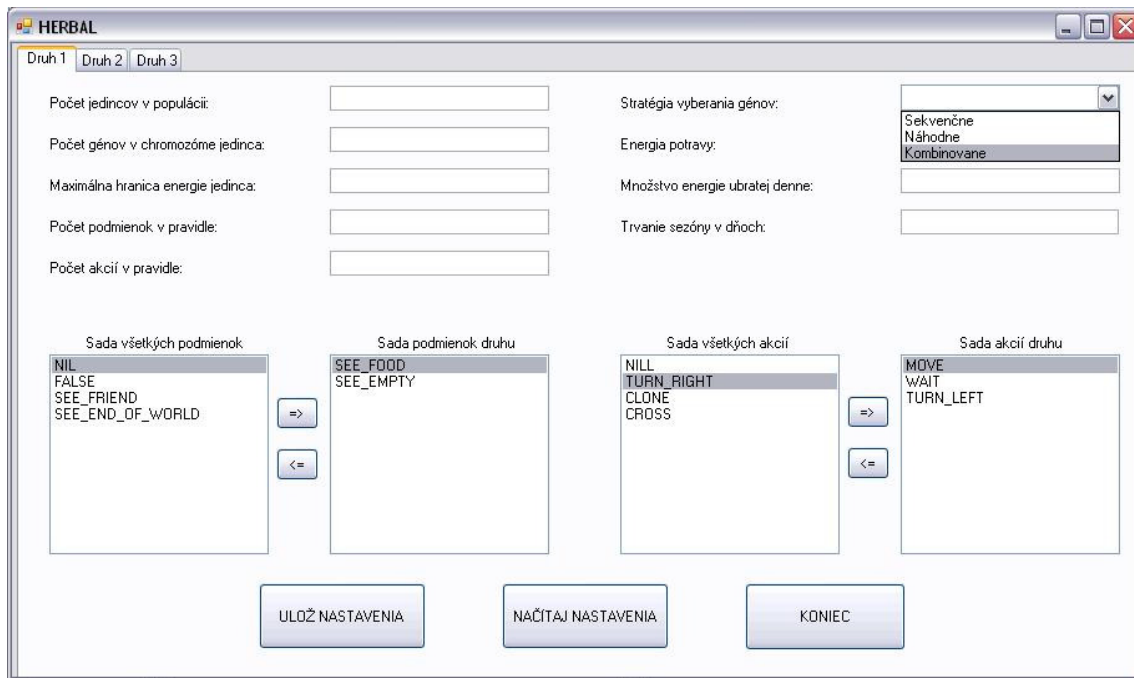
Obr. 5 – Diagram tried vizualizačnej časti

3.2.2 GUI

Aplikácia je navrhnutá ako klasická oknová aplikácia v prostredí Windows, využívajúca pre grafické používateľské rozhranie známe ovládacie prvky z tohto prostredia. Jednoduchý návrh obrazovky pre nastavenie počiatočných parametrov sveta je možné vidieť na obrázku (Obr. 6).

3.2.3 Vizualizácia sveta

Na vizualizáciu sveta sú potrebné parametre, ktoré boli popísané vyššie. Na obrázku (Obr. 7) je možné vidieť zjednodušený návrh takejto vizualizácie. Každé políčko má svoju textúru, vďaka ktorej je možné rozoznať, či sa na políčku nachádza, alebo nenachádza stena. Objekty pohybujúce sa vo svete je možné označiť, pri tomto označení sa im zmení textúra a zobrazia sa o nich jednotlivé informácie. Vizualizáciu je možné kedykoľvek zastaviť a znova spustiť.



Obr. 6 – GUI nastavenia začiatkových parametrov sveta



Obr. 7 – Jednoduchý návrh vizualizácie sveta

Dokumentácia riadenia projektu

Vedúci projektu	Ing. Pavel Bartoš
Členovia tímu č. 19	Bc. Miloš Auder
	Bc. Andrej Belica
	Bc. Lukáš Ďurčák
	Bc. Miroslav Mikuláš
	Bc. Martin Paššák
	Bc. Ján Romaňák

4 Úvod k riadeniu projektu

V tejto časti dokumentácie uvedieme informácie týkajúce sa riadenia projektu HERBAL v tíme č.:19 – Herbovia. Na začiatku zverejňujeme našu ponuku, na základe ktorej sme tento projekt pre svoj tím získali v nezmenenej podobe. V nasledujúcej kapitole sú uvedené popisy a hodnotenia napĺňania projektového plánu v jednotlivých etapách kontrolovania progresu projektu. Ďalej sa nachádzajú popisy dlhodobých a dočasných úloh jednotlivých členov tímu spolu s ich odôvodneniami. Nasledovnú časť tejto dokumentácie tvorí zoznam zápisov zo stretnutí tímu so zadávateľom projektu Ing. Bartošom. Potom sa v dokumentácii nachádzajú metodiky, ktoré sme v tíme pre vývoj tohto projektu schválili. V ďalšej kapitole sú popísané procesy manažmentu verzií, ktoré používame pri tvorbe projektu a jeho dokumentácie. Záver tejto časti dokumentácie tvorí zoznam preberacích protokolov.

5 Ponuka

Tu uvádzame ponuky na tímový projekt tak ako sme ich vypracovali na začiatku zimného semestra akademického roka 2010/2011. Ponuky sú uvedené s použitím ich pôvodného formátovania (s miernymi úpravami). Okrem ponuky na nám nakoniec pridelenú tému „Evolučný simulátor umelého života založený na heuristických pravidlách“ sa tu nachádza aj alternatívna ponuka pre tému „Portál pre časopis ACM“.

PONUKY - TÍM 19

O TÍME

Sme tím šiestich ľudí, z toho piati sme absolventi bakalárskeho štúdia, študijného programu Informatika na Fakulte informatiky a informačných technológií Slovenskej technickej univerzity v Bratislave. Šiesty člen je absolventom bakalárskeho štúdia na Fakulte riadenia a informatiky Žilinskej univerzity. Sme zohraný tím, keďže viacerí sa poznáme už niekoľko rokov a spolupracovali sme aj na množstve projektov.

Miloš Auder – počas štúdia sa naučil pracovať s viacerými programovacími jazykmi a technológiami ako napr. C, C++, C#.NET, MySQL, PHP, Matlab a XML. Vo svojej bakalárskej práci sa venoval Optimalizácii inteligentných dopravných systémov v prostredí Matlab.

Andrej Belica – má pokročilé programátorské zručnosti v jazykoch C, C++, Java, PHP a Javascript. Zaujíma sa o najnovšie technológie v oblasti IT. V bakalárskej práci implementoval modul do systému Moodle, ktorý vylepšoval podporu vkladania študentských zadaní.

Lukáš Ďurčák – v súčasnosti pracuje na vývoji webových systémov v reklamnej agentúre. Aktívne sa venuje programovaniu v PHP a okrem toho ovláda aj C++, C#, Javu a má bohaté skúsenosti s prácou s databázami. V bakalárskej práci implementoval svoj vlastný webový redakčný systém (CMS).

Miroslav Mikuláš – má skúsenosti v práci s C, C++, PHP, MySQL a Java. Zaujíma sa o moderné trendy v používateľských rozhraniach a témou jeho bakalárskej práce boli Grafické používateľské rozhrania v 3D prostredí.

Martin Paššák – má skúsenosti v práci s C, C++, PHP, Javascript, MySQL. Bakalársku tému vypracoval na tému Efektívnych údajových reprezentácií reálnych sietí. V rámci tímu má najlepšie skúsenosti s databázami.

Ján Romaňák – medzi jeho záľuby patrí programovanie v C, Java, Lisp, Prolog a PHP. Téma jeho bakalárskej práce bola „Game of Life“, v ktorej skúmal štruktúry umelého života.

Evolučný simulátor umelého života založený na heuristických pravidlách

Motivácia

S touto témou máme skúsenosti už z bakalárskeho štúdia, z predmetu Umelá inteligencia. Zaujala nás až natoľko, že na inžinierskom štúdiu sme si zapísali niekoľko ďalších predmetov súvisiacich s touto problematikou, ako napríklad Strojové učenie (4 členovia) a Evolučné algoritmy (5 členov).

Prostredníctvom tímového projektu by sme chceli prehĺbiť a rozšíriť naše vedomosti a znalosti v oblasti umelej inteligencie a evolučných algoritmov. Sme presvedčení o tom, že náš záujem o túto tému bude mať za výsledok náš aktívny prístup pri práci na projekte. Fascinuje nás široké spektrum možností využitia simulátora a voľnosť pri jeho tvorbe. Veríme, že charakter témy udrží náš záujem počas celého procesu vývoja.

Dúfame, že si budeme môcť zlepšiť naše schopnosti práce v tíme práve pri takejto pútavej téme.

Našou prácou by sme chceli skúmať, do akej miery sme schopní priblížiť sa k mechanizmu evolúcie v reálnom svete. Radi by sme preskúmali vplyv rôznych faktorov na evolúciu. Ide napríklad o vplyv génovej diverzity, rôznych atribútov prostredia a organizmov, faktor náhody a iné.

Tešíme sa na zaujímavú prácu, ktorá má dobrý potenciál vyústiť do zaujímavých výsledkov.

Koncepcia riešenia

Simulátor plánujeme vytvoriť prostredníctvom objektovo-orientovaného prístupu, pretože veríme že objektová reprezentácia dát z tejto oblasti je tou najvhodnejšou a umožní efektívnu a prehľadnú implementáciu. Efektivitu chceme zabezpečiť rozumným použitím multithreadingu a dobrej reprezentácie dát. Multithreading umožní sledovanie evolúcie vo väčšom merítku a zabezpečí reálnejšie správanie sa organizmov v simulácii.

Naším zámerom je venovať zvýšenú pozornosť vizualizácii, aby sme vytvorili príjemný a ľahko používateľný produkt, ktorý by potenciálneho používateľa zaujal. Prostredníctvom používateľského rozhrania náš produkt poskytne jednoduchú možnosť konfigurovať veľké množstvo parametrov.

Následky zmien týchto parametrov sa budú dať jednoducho sledovať v štatistikách, ktoré bude program sprístupňovať. Chceme sa sústrediť na sledovanie vplyvu:

- evolučných mechanizmov (mutácie, kríženie...)
- vlastnosti prostredia
- náhodných zmien (simulácia prírodných katastrof...)
- schopnosti prispôbiť sa novým podmienkam
- veľkosti a rôznorodosti populácie
- hierarchie organizmov

Plánujeme vytvoriť základné riešenie s niekoľkými druhmi organizmov (pasívne organizmy, zberači, lovci). Zabezpečíme však aj možnosť rozšíriť produkt, napríklad o nové počiatkové organizmy alebo vlastnosti sveta. Všetkým objektom chceme prideliť atribúty pomocou návrhového vzoru Composite, ktorý zabezpečí v podstate neobmedzenú rozšíriteľnosť vlastností.

Portál pre časopis

Motivácia

Po získaní skúseností s webovými technológiami (PHP, Javascript, Ajax) sme zistili, že takýto druh práce nás zaujíma a naplňuje. Veľmi nás láka možnosť spolupracovať s medzinárodne uznávanou organizáciou ACM. Tvorba takéhoto webového portálu zároveň predstavuje veľmi zaujímavú výzvu, ktorá nám umožní uplatniť náš tvorivý potenciál. Veríme, že prácou na tomto projekte získame pre nás dôležité a do praxe užitočné referencie a skúsenosti. Ak by sme dobre implementovali používateľskú funkcionality (používateľské kontá, možnosť pridávania komentárov...), veríme že by sa časom vybudovala zaujímavá komunita.

Koncepcia riešenia

V našom tíme niet člena ktorý by nemal skúsenosti či už s návrhom rozhrania, vývojom webových aplikácií alebo databázovými technológiami. To bol jeden z dôvodov prečo sme sa rozhodli pre jazyk PHP a MySQL. Naším cieľom je zabezpečiť jednoduché vkladanie článkov do centrálného úložiska, kde budú čakať na schválenie. Články budeme kategorizovať a zabezpečíme automatické

prideľovanie článkov posudzovateľom. Chceme čitateľom poskytnúť možnosť spätnej väzby, napr. v podobe ohodnocovania a komentovania článkov čitateľmi.

Samozrejmosťou je rozdelenie systému na administrátorskú a používateľskú časť s možnosťou vytvárať rôzne stupne používateľských práv.

Pri tvorbe systému chceme využiť návrhový vzor MVC ktorý nám umožni pracovať na jednotlivých komponentoch systému oddelene. Naším cieľom je tvoriť systém modulárne, okrem modulov umožňujúcich základné fungovanie chceme vytvoriť aj rôzne doplnkové moduly, ktoré budú poskytovať doplnkové služby. Web stránku chceme navrhnuť tak aby využívala technológie web 2.0. Súčasťou nášho riešenia bude aj vytvorenie kvalitného API tak, aby bolo možné systém v budúcnosti jednoducho rozširovať o nové moduly/funkcie.

Príloha A – Zoradenie tém podľa záujmu

- Evolučný simulátor umelého života založený na heuristických pravidlách
- Portál pre časopis
- Crowdsourcing verejných dát
- RoboCup tretí rozmer
- Tréner mentálnych schopností
- Prispôsobiteľný Widget
- Interaktívna vizualizácia grafových štruktúr v 3D priestore
- Správa študentských projektov na fakulte
- Objektové úložisko dát
- Dizajn s použitím obohatenej reality
- Virtuálna FIIT
- Model používateľa pre jeho identifikáciu
- Simulated Car Racing Competition 2011
- Platforma pre realizovanie transakcií prostredníctvom mobilných zariadení

- Tvorba rozvrhov
- Vyhľadavanie a sprístupnenie citácií
- Adaptívny proxy server
- 3D grafická podpora vyhľadavania znalostí v dokumentoch

Príloha B – Rozvrh

	7 ⁰⁰ - 7 ⁵⁰	8 ⁰⁰ - 8 ⁵⁰	9 ⁰⁰ - 9 ⁵⁰	10 ⁰⁰ - 10 ⁵⁰	11 ⁰⁰ - 11 ⁵⁰	12 ⁰⁰ - 12 ⁵⁰	13 ⁰⁰ - 13 ⁵⁰	14 ⁰⁰ - 14 ⁵⁰	15 ⁰⁰ - 15 ⁵⁰	16 ⁰⁰ - 16 ⁵⁰	17 ⁰⁰ - 17 ⁵⁰	18 ⁰⁰ - 18 ⁵⁰	19 ⁰⁰ - 19 ⁵⁰	20 ⁰⁰ - 20 ⁵⁰
PO		VI			VI	PDBT		PDBT / SU		TP I		VSS / VIS		
				3		3	1,4,5		3 / 1,2,5,6		1,2,3,4,5,6		1,2,4,5,6 / 3	
UT	KÓD								MSI		MSI		MSI	
		4								1,2,3,4,5,6		1,2,3,4,5,6		1,2,3,4,5,6
ST			SU		AOVS / TV					KÓD				
				1,2,5,6		2,6 / 5					4			
ŠT			TV						ASS	AIS				
				3						1,2,4,5,6		3		
PI			PDBT				AOVS							
				1,3,4,5				2,6						

- 1 - Auder Miloš
- 2 - Belica Andrej
- 3 - Ďurčák Lukáš
- 4 - Mikuláš Miroslav
- 5 - Paššák Martin
- 6 - Romaňák Ján

	prednáška
	cvičenie
	chodenie domov ☺

Mailový kontakt: milos.auder@gmail.com

6 Projektový plán

V tabuľke (Tab. 1) je hrubý plán na zimný semester, ktorý bol vytvorený 7.10.2010.

Tab. 1 – Hrubý plán

Úloha	Začiatok	Koniec
Dokončenie stránky tímu	7.10.2010	18.10.2010
Vytvorenie špecifikácie požiadaviek	7.10.2010	21.10.2010
Vytvorenie analýzy	7.10.2010	21.10.2010
Vytvorenie návrhu	7.10.2010	4.11.2010
Osvojenie si vizualizácie v prostredí C#	7.10.2010	4.11.2010
Vytvorenie prototypu	4.11.2010	14.12.2010

V tabuľke (Tab. 2) je spresnený plán vytvorený 3.11.2010. Jedná sa hlavne o spresnenie časti „Vytvorenie prototypu“ hrubého plánu, keďže ostatné časti hrubého plánu boli v čase vytvárania aktuálneho plánu splnené.

Tab. 2 - Aktuálny plán

Úloha	Začiatok	Koniec
Implementácia sveta	4.11.2010	11.11.2010
Implementácia riadenia	4.11.2010	11.11.2010
Implementácia GUI	4.11.2010	11.11.2010
Implementácia reportovania	4.11.2010	11.11.2010
Implementácia druhov a organizmov	11.11.2010	18.11.2010
Implementácia DNA a génov	11.11.2010	18.11.2010
Implementácia vizualizácie	11.11.2010	25.11.2010
Implementácia akcií a podmienok	18.11.2010	25.11.2010
Testovanie, opravy chýb a optimalizácia	25.11.2010	9.12.2010
Tvorba dokumentácie	25.11.2010	9.12.2010

7 Úlohy členov v tíme

V treťom týždni semestra si náš tím určil hlavné pozície a úlohy pre jednotlivých členov. Tieto by mali byť v platnosti minimálne do konca zimného semestra. Sú to:

- Vedúci tímu - Martin Paššák
- Manažér vývoja - Andrej Belica
- Manažér podpory - Lukáš Ďurčák
- Manažér kvality - Miloš Auder
- Zodpovedný za dokumentáciu - Miroslav Mikuláš
- Zodpovedný za analýzu a návrh - Ján Romaňák

V ďalších týždňoch, keď už sa začínala plánovať práca na prototype, sa náš tím rozdelil na dve menšie časti. Prvá časť tímu, ktorá má na starosti implementáciu jadra aplikácie, má nasledujúce zloženie:

- Andrej Belica
- Martin Paššák
- Ján Romaňák

Druhá časť tímu, ktorá sa zaoberá vizualizáciou systému a grafickým používateľským rozhraním, má nasledujúce zloženie:

- Miloš Auder
- Lukáš Ďurčák
- Miroslav Mikuláš

8 Záznamy zo stretnutí

V tejto kapitole uvádzame zápisy zo stretnutí tímu so zadávateľom projektu Ing. Bartošom. Zápisy sú uvádzané s použitím ich originálneho formátovania

Zápis z 1. stretnutia tímu č.:19

Dátum: 29. 9. 2010

Miestnosť: Softvérové štúdio FIIT STU

Vypracoval: Bc. Ján Romaňák

Prítomní:

Vedúci: Ing. Pavel Bartoš

Členovia tímu: Bc. Miloš Auder, Bc. Andrej Belica, Bc. Lukáš Ďurčák, Bc. Miroslav Mikuláš, Bc. Martin Paššák, Bc. Ján Romaňák

Vyhodnotenie úloh z predchádzajúceho stretnutia:

Žiadne úlohy neboli stanovené.

Obsah stretnutia:

Diskusia o základných princípoch modelu sveta:

- gény vo forme pravidiel 'if – then'
 - minimálna sada podmienok: always true/false, ...
 - minimálna sada akcií: clone, cross, turn, move, die, noop (wait), nill
 - akcie majú so sebou spojené rôzne veľké výdaje energie
 - Podmienka/akcia môže pozostávať z viacerých elementárnych podmienok/akcií
- spôsob výberu pravidiel, ktoré sa uplatnia v danom kroku: náhodný, sekvenčný, ...
- sledovanie štatistík ako napr.: veľkosť, vek, priemerná energia populácie
- voľba dobrého random generátora
 - zaznamenávanie random seeds – aby bolo možné zopakovať konkrétne simulácie

Úlohy do ďalšieho stretnutia:

Číslo úlohy	Dátum zadania	Termín na splnenie	Úloha	Zodpovedná osoba
1.1	29. 9. 2010	7. 10. 2010	Premyslieť si implementačné prostredie	celý tím
1.2	29. 9. 2010	7. 10. 2010	Určiť role v tíme	celý tím

Zápis z 2. stretnutia tímu č.19:

Dátum: 7. 10. 2010

Miestnosť: Softvérové štúdio FIIT STU

Vypracoval: Bc. Miloš Auder

Prítomní:

Vedúci: Ing. Pavel Bartoš

Členovia tímu: Bc. Miloš Auder, Bc. Andrej Belica, Bc. Lukáš Ďurčák, Bc. Miroslav Mikuláš, Bc. Martin Paššák, Bc. Ján Romaňák

Vyhodnotenie úloh z predchádzajúceho stretnutia:

Číslo úlohy	Dátum zadania	Termín na splnenie	Dátum splnenia	Úloha	Zodpovedná osoba	Stav
1.1	29. 9. 2010	7. 10. 2010	5. 10. 2010	Premyslieť si implementačné prostredie	Celý tím	Splnená
1.2	29. 9. 2010	7. 10. 2010	5. 10. 2010	Určiť role v tíme	Celý tím	Splnená

Obsah stretnutia:

Prezentovanie rozdelenia rolí:

- Martin Paššák – vedúci tímu
- Miloš Auder – manažér kvality
- Miroslav Mikuláš – dokumentarista
- Andrej Belica – manažér vývoja
- Lukáš Ďurčák – manažér podpory
- Ján Romaňák

Odprezentovanie implementačného prostredia – C#

Prediskutovanie otázok ktoré vyplynuli z minulého stretnutia:

- Hranica maximálnej energie – môžeme neskôr doplniť ako parameter
- Svet potravy – rast, rozširovanie – na doplnenie neskôr
- Zjedenie potravy aj ako akcia – na doplnenie neskôr
- Čas – po tiku (real-time) – ak by sa ukázalo že je to pomalé a pod. malo by sa dať vrátiť k priebehu času po dňoch

- Viac druhov energie – na začiatok by mala byť iba jedna (vitalita), neskôr možnosť doplniť iný typ energie
- Ak máme 2 druhy jedincov, ako ich na začiatku rozmiestnime? Ako ich budeme vyberať na vykonávanie akcií? Malo by to byť náhodne, ale bolo by zaujímavé ich rozmiestniť manuálne.
- Do budúcnosti porozmýšľať
 - vizualizácia veľkého sveta (posúvanie, ZOOM ...)
 - aké štatistiky by sa mali dať sledovať (priemerný vek dožitia, priemerný počet potomkov ...)
 - iný svet ako štvorec?

Úlohy do ďalšieho stretnutia:

Číslo úlohy	Dátum zadania	Termín na splnenie	Úloha	Zodpovedná osoba
2.1	7. 10. 2010	14. 10. 2010	Vytvoriť plán	Andrej Belica
2.2	7. 10. 2010	18. 10. 2010	Vytvoriť web stránku tímu	Lukáš Ďurčák
2.3	7. 10. 2010	4. 11. 2010	Vytvoriť analýzu, špecifikáciu, návrh	celý tím

Zápis z 3. Stretnutia tímu č. 19

Dátum: 14. 10. 2010

Miestnosť: Softvérové štúdio FIIT STU

Vypracoval: Bc. Andrej Belica

Prítomní:

Vedúci: Ing. Pavel Bartoš

Členovia tímu: Bc. Miloš Auder, Bc. Andrej Belica, Bc. Lukáš Ďurčák, Bc. Miroslav Mikuláš, Bc. Martin Paššák, Bc. Ján Romaňák

Vyhodnotenie predchádzajúcich úloh:

Číslo úlohy	Dátum zadania	Termín na splnenie	Dátum splnenia	Úloha	Zodpovedná osoba	Stav
2.1	7.10.2010	14.10.2010	13.10.2010	Vytvoriť plán	Andrej Belica	Splnená
2.2	7.10.2010	18.10.2010		Vytvoriť web stránku tímu	Lukáš Ďurčák	Rozpracovaná
2.3.1	7.10.2010	21.10.2010		Vytvoriť špecifikáciu požiadaviek	Miroslav Mikuláš, Miloš Auder	Nezačatá
2.3.2	7.10.2010	21.10.2010		Vytvoriť analýzu	Ján Romaňák, Lukáš Ďurčák	Nezačatá
2.3.3	7.10.2010	4.11.2010		Vytvoriť návrh	Andrej Belica, Martin Paššák, Ján Romaňák	Nezačatá

Obsah stretnutia:

Prezentovanie nainštalovaného softvéru pre podporu riadenia projektu – dotproject.

Prezentovanie plánu – rozdelenie úloh.

Začiatok práce na špecifikácii požiadaviek a návrhu – podľa rozdelenia úloh.

Špecifikácia požiadaviek – napísaná časť textu.

Návrh – neformálne nakreslená časť diagramu tried.

Aktuálne úlohy:

Číslo úlohy	Dátum zadania	Termín na splnenie	Úloha	Zodpovedná osoba
2.2	7.10.2010	18.10.2010	Vytvoriť web stránku tímu	Lukáš Ďurčák
2.3.1	7.10.2010	21.10.2010	Vytvoriť špecifikáciu požiadaviek	Miroslav Mikuláš, Miloš Auder
2.3.2	7.10.2010	21.10.2010	Vytvoriť analýzu	Ján Romaňák, Lukáš Ďurčák
2.3.3	7.10.2010	4.11.2010	Vytvoriť návrh	Andrej Belica, Martin Paššák, Ján Romaňák

Zápis zo 4. Stretnutia tímu č. 19

Dátum: 21. 10. 2010

Miestnosť: Softvérové štúdio FIIT STU

Vypracoval: Bc. Martin Paššák

Prítomní:

Vedúci: Ing. Pavel Bartoš

Členovia tímu: Bc. Miloš Auder, Bc. Andrej Belica, Bc. Lukáš Ďurčák, Bc. Miroslav Mikuláš, Bc. Martin Paššák, Bc. Ján Romaňák

Vyhodnotenie predchádzajúcich úloh:

Obsah stretnutia:

- [I] Prezentovanie zápisnice zo stretnutia č.3.
- [R] Pozmenenie formátu zápisníc(pridanie tagov označujúcich typ činnosti na stretnutí [I] – Informácia, [R] – Rozhodnutie, [U] – Úloha, zrušenie tabuľky zo splnenými úlohami).
- [I] Prezentovanie analýzy.
- [U] Rozšíriť analýzu o rozdiel medzi umelou inteligenciou a umelým životom [2.3.2].
- [U] Vypracovať hypotézy pre analýzu [4.3.1].
- [I] Prezentovanie špecifikácie požiadaviek.
- [I] Diskusia k meraniu úspešnosti populácie.
- [R] Rozšírenie funkcionality o ukladanie dát a štatistík do súboru offline.
- [R] Používať iba pozitívne testovanie pri senzoroach.
- [R] Použiť fuzzy množiny pre číselné veličiny.
- [R] Odstránenie viacerých druhov energie zo špecifikácie.
- [I] Diskusia k atribútu pohlavia, jeho vplyvu na vývoj populácie a potrebných akciách.
- [I] Diskusia k spôsobu tímového programovania.
- [I] Diskusia k vizuálnej stránke (grafické efekty, zvuky, náročnosť).
- [R] Písať verzie, dátumy a autorov aj do dokumentov.
- [U] Vypracovanie prihlášky na TPCup [4.4.1].
- [R] Pridať do dokumentácie slovník na definovanie pojmov.

Vyhodnotenie úloh:

Číslo úlohy	Dátum zadania	Termín na splnenie	Úloha	Zodpovedná osoba	Stav
2.3.1	7.10.2010	21.10.2010	Vytvoriť špecifikáciu požiadaviek	Miroslav Mikuláš, Miloš Auder	Splnená
2.3.2	7.10.2010	21.10.2010	Vytvoriť analýzu	Ján Romaňák, Lukáš Ďurčák	80%
2.3.3	7.10.2010	4.11.2010	Vytvoriť návrh	Andrej Belica, Martin Paššák, Ján Romaňák	10%
2.3.4	7.10.2010	4.11.2010	vizualizácia v C#	Miloš Auder, Lukáš Ďurčák, Miroslav Mikuláš	10%
4.3.1	21.10.2010	28.10.2010	Vytvorenie hypotéz pre analýzu	Andrej Belica, Martin Paššák, Ján Romaňák, Miloš Auder, Lukáš Ďurčák, Miroslav Mikuláš	Nezačatá
4.4.1	21.10.2010	24.11.2010	Vytvorenie prihlášky na TPcup	-	Nezačatá

Zápis z 5. Stretnutia tímu č. 19

Dátum: 28. 10. 2010

Miestnosť: Softvérové štúdio FIIT STU

Vypracoval: Bc. Miroslav Mikuláš

Prítomní:

Vedúci: Ing. Pavel Bartoš

Členovia tímu: Bc. Miloš Auder, Bc. Andrej Belica, Bc. Lukáš Ďurčák, Bc. Miroslav Mikuláš, Bc. Martin Paššák, Bc. Ján Romaňák

Obsah stretnutia:

- [I] Prezentovanie zápisnice zo stretnutia č.3.
- [I] Zvolený nástroj na verziovanie systému - SVN.
- [I] Prezentácia stránky sourceforge.net.
- [I] Prezentovanie finálnej verzie analýzy.
- [I] Prezentovanie hypotéz pre analýzu.
- [R] Všetky hypotézy v analýze by malo byť možné overiť už v prototype, ktorý sa odovzdáva v zimnom semestri.
- [I] Prezentovanie návrhu jadra systému.
- [I] Prezentovanie návrhu grafického používateľského rozhrania.
- [I] Ukážka vytvoreného GUI pre nastavenie počiatočného stavu sveta.
- [I] Ukážka jednoduchej vizualizácie, demonštrácia možností XNA - zvoleného nástroja pre vizualizáciu.
- [R] Svet vo vizualizácii sa bude dať približovať a oddiaľovať aj v 2d prototype - použitím zmeny rozmerov políček.
- [R] V dokumentácii pridať autorov jednotlivých častí do hlavičky.
- [R] Strany v dokumentácii číslovať podľa jednotlivých kapitol (napr. 1-1, 1-2..)
- [R] Do špecifikácie doplniť podmienku see_friend o podmienku, kedy herb vidí aj herba iného druhu.
- [R] Do dokumentácie dať finálnu verziu špecifikácie.

Vyhodnotenie úloh:

Číslo úlohy	Dátum zadania	Termín na splnenie	Úloha	Zodpovedná osoba	Stav
2.3.2	7.10.2010	21.10.2010	Vytvoriť analýzu	Ján Romaňák, Lukáš Ďurčák	Splnená
2.3.3	7.10.2010	4.11.2010	Vytvoriť návrh	Andrej Belica, Martin Paššák, Ján Romaňák	30%
2.3.4	7.10.2010	4.11.2010	vizualizácia v C#	Miloš Auder, Lukáš Ďurčák, Miroslav Mikuláš	60%
4.3.1	21.10.2010	28.10.2010	Vytvorenie hypotéz pre analýzu	Andrej Belica, Martin Paššák, Ján Romaňák, Miloš Auder, Lukáš Ďurčák, Miroslav Mikuláš	Splnená
4.4.1	21.10.2010	24.11.2010	Vytvorenie prihlášky na TPcup	Miloš Auder, Martin Paššák	Nezačatá

9 Metodiky

9.1 Metodika pre písanie zdrojového kódu

Všeobecné pravidlá pre písanie zdrojového programu v prostredí MS Visual Studio 2008, v jazyku C#:

- Jednoslovné premenné začať malým písmenom, napr.:

```
int pocet;
```

- Viacslovné premenné začať malým písmenom, a každé ďalšie slovo začať veľkým písmenom, napr.:

```
int pocetExternychZamestnancov;
```

- Názvy tried a metód začínať veľkým písmenom, začiatkové a koncové zátvorky budú na samostatných riadkoch, pričom riadky kódu vnútri metódy budú odsadené na veľkosť tabulátora, napr.:

```
void NapisAhoj (string retazec)
{
    Console.WriteLine (retazec);
}

class Herb
{
    public Herb ()
    {
    }
}
```

- Nepoužívať skratky pri názvoch premenných, jedinou výnimkou sú premenné používané v cykloch, napr.:

```
for (int i = 0; i < pocet; i++)
{
    Console.WriteLine (i);
}
```

- Komentáre písať na tej istej úrovni ako je kód ktorý popisujú, napr.:

```

// Adresa klienta
string adresa;

void PosliEmail (string email)
{
    // Posle email na adresu...

    // ...
}

```

- Členské premenné deklarovať ako *private*, manipulovať s nimi pomocou *public/protected* properties, napr.:

```

private double sekundy;
const int hodinaSekundy = 3600;

public double Hodiny
{
    get { return sekundy / hodinaSekundy; }
    set { sekundy = value * hodinaSekundy; }
}

```

- Nepoužívať v kóde tzv. „magické“ čísla a reťazce, ale deklarovať ich ako konštanty, viď obrázok vyššie.