



## **Programová dokumentácia k projektu agents2007 do predmetov CPP a IPP**

**Simulátor agentného systému**

Autori: Martin Šūs, [xsusma00@stud.fit.vutbr.cz](mailto:xsusma00@stud.fit.vutbr.cz)  
Dávid Smejkl, [xsmejk13@stud.fit.vutbr.cz](mailto:xsmejk13@stud.fit.vutbr.cz)  
Erik Nagy, [xnagye00@stud.fit.vutbr.cz](mailto:xnagye00@stud.fit.vutbr.cz)  
Andrej Trnkóci, [xtrnko00@stud.fit.vutbr.cz](mailto:xtrnko00@stud.fit.vutbr.cz)  
Dátum: 29. 4. 2007

# Obsah

<b>1. Úvod</b>	3
<b>2. Triedy</b>	3
2.1 Trieda SCENA	3
2.2 Trieda AGENT	5
<b>3. Interpret Lua</b>	7
<b>4. XML</b>	8

## ***1. Úvod***

Programová dokumentácia obsahuje stručný a ucelený popis tried ich metód a dát použitých pri projekte. Dokumentácia slúži na orientáciu sa v kóde programu agents2007, ktorý je grafickou aplikáciou využívajúcou GUI cross-platform toolkit wxWidgets.

## 2. Triedy

V programe je hlavnou riadiacou triedou SCENA, ktorá sa stará o priebeh simulácie a je spojením s grafickým prostredím programu. Ďalšou triedou je AGENT, ktorá je zodpovedá za informácie o agentoch ktorých obsahuje scéna. Pri kroku agenta sa využíva interpret Lua, ktorá vykonáva chovanie agenta programované užívateľom.

### 2.1 Trieda SCENA

Trieda SCENA zaist'uje priebeh simulácie a informácie o aktuálnom aj predošlom stave scény. K tomu jej slúžia nasledujúce dátové typy a štruktúry:

**pole** – je jednorozmerné pole štruktúr **SPozicia** (obsahuje dva prvky typu integer), ktoré má rovnaký počet položiek ako scéna (súradnice sú prepočítavané).

**pocetkol** – integer s informáciou o počte kôl, ktoré scéna vykonala.

**agents** – zoznam ukazateľou na agentov v scéne. Agenti sú zoradený od najstaršieho po najmladšieho.

**rozmerx** a **rozmary** – integere s informáciou o rozmeroch scény.

**obsah** – je výčtový typ enum, ktorého položky slúžia ako informácia o stave pozície v poli scény.

Na prácu z týmito dátami slúžia nasledovné metódy:

**SCENA(int y, int x)** – je konštruktorom novej scény.

**void odlozscenu()** - odloží informácie o scéne pred novým kolom, pre budúce porovnanie rozdielov v novej a starej scéne. Tu sa využíva štruktúra **SPozicia** ktorú obsahujú položky **pole**. Táto vlastnosť je použitá pri vykresľovaní scény.

**void kolo()** - scéna volá agentov zo zoznamu, aby postupne spravili krok. Agenti prepisujú v kroku informáciu o obsahu políček v scéne, cez ktoré prechádzajú.

**int cojetam(int x, int y),**

**int cotambolo(int x, int y)** - vráti integer z enumu **obsah**, ktorí určuje čo je v aktuálnej a v predošlej scéne na pozícii x, y.

**void zmazpoziciu(int x, int y)** – uvoľní pozíciu v scéne, pokiaľ bol na pozícii agent, zmaže ho zo zoznamu agentov.

**int vlozstenu(int x, int y),**  
**int vlozagenta(AGENT agent),**  
**int vlozpredmet(int x, int y)** -metódy slúžia na vkladanie objektov (stena, agent a predmet) na pozície v scéne, a pri vložení agenta zanesú ukazateľ naň aj do zoznamu agentov, kde ho pridajú na koniec. Funkcie v prípade nesprávneho použitia vracajú hodnotu 1.

**int nacitajagenta(AGENT agent)** – načítanie agenta sa využíva pri načítavaní uloženej scény. V prípade chyby vracia hodnotu 1.

**void presunagenta(int odkialx, int odkialy, int kamx, int kamy, char predmet),**  
**void presunagentaglob(int odkialx, int odkialy, int kamx, int kamy)** – premiestnenie agenta v scéne pri kroku simulácie a pri editácii scény.

**void zmensmer(int x, int y, char novysmer)** – zmení smer agenta v scéne na danej pozícii a aj v agentovi samotnom.

**char odovzdajpredmet(int x, int y)** – Vracia premennú char určujúcu predmet ktorý zo scény zdvihol agent na danej pozícii. Tento predmet si agent uloží.

**void prijmiPredmet(int x, int y)** – scéna prijme predmet od agenta na pozícii.

## 2.2 Trieda AGENT

Trieda AGENT predstavuje informácie o agentovi, ktorý je na scéne. Obsahuje dáta ktoré určujú presné vlastnosti každého agenta.

**posx, posy** – identifikátory typu integer, určujú pozíciu na scéne. Podľa nich je možné identifikovať agenta v zozname agentov scény.

**smerpohybu** – char, s informáciou (**U, D, L, R**) o smere agenta.

**predmet** – informácia typu char, o predmete, ktorý nesie agent, alebo je agent prázdny.

**scena** – odkaz na scénu, v ktorej je agent.

**skript** – string s menom skriptu, podľa ktorého sa agent správa.

**L** – odkaz na premennú **lua State**. Agent má cez túto premennú prístup k funkciám zaregistrovaným interpretom Lua.

Metódy triedy AGENT:

**AGRNT(int y, int x, char smer, char teleso, SCENA \*Scena, string Skript)** – je konštruktorom nového agenta.

**void krok()** - agent na seba volá interpret Lua, ktorého registrované funkcie ovládajú agenta podľa uloženého skriptu.

**int vratpoziciux()**,

**int vratpoziciuy()**,

**int vratsmer()** - návratové hodnoty udávajú informáciu o smere a pozícii agenta.

### ***3. Interpret Lua***

Ako interpret skriptov riadiacich agentov bol využitý jazyk LUA. K projektu sa pri preklade prilinkuje knihovna LUA 5.0.3, do zdrojových súboru. Hlavičkové súbory nevyhnutné pre prácu s týmto interpretom - "lua.h" "lualib.h" "lauxlib.h" sú pridané v súbore "engine.h". Tu je využitý fakt že LUA dokáže volať funkcie programovacieho jazyka c. Aby bolo možné dané funkcie takto používať, je nutné ich najprv v lue zaregistrovať. Funkcie sa registrujú do lui pomocou funkcie lua\_register. Táto funkcia má tri parametre – prví parameter typu lua\_state označuje stav interpretácie daného skriptu, druhý typu char\* je meno funkcie s akým sa daná funkcia bude volať v lue a tretí je meno funkcie v jazyku c/c++ ktorá sa má zavolať. Funkcie lua\_register sa volajú v konštruktoze triedy agent. Funkcie volané z lui sú všetky globálne, dôvod prečo je to takto vyriešené je že lua nevie volať metódy. Aby tieto funkcie boli schopné zistiť ktorého agenta majú editovať, je použitá globálna premenná GlobAgent typu ukazateľ na agent, do ktorej sa ukladá vždy agent ktorý práve vykonáva svoj krok.

## 4.XML

Ukladanie nastavení skriptov a scén je realizované vo formáte XML. Pre zapisovanie a čítanie z XML súborou sa používa XML parser TinyXML. Zdrojové kódy TinyXML sú pridané do projektu a prekladajú sa spolu s programom.

Formát XML súboru s lua skriptom:

```
<skript>samotný skript</skript>
```

Formát uloženej scény:

```
<SCENA x="rozmer x" y="rozmer y">
  <SURADNICE>
    <X:Y x="0" y="0" content="1" />
    <X:Y x="0" y="1" content="1" />
    .....ďalšie súradnice (content - čo sa na daných súradniciach nachádza)
  </SURADNICE>
  <AGENTS>
    <AGENT x="3" y="3" smer="82" predmet="80" skript="skript.xml" />
    <AGENT x="5" y="6" smer="85" predmet="45" skript="skript.xml" />
    ...ďalší agenti v scéne
  </AGENTS>
</SCENA>
```