Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Ilkovičova 3, 812 19 Bratislava

RoboCup 3D

Používateľská a systémová príručka

Tímový projekt

Autori: Bc. Peter Nosko
Bc. Dušan Rodina
Bc. Daniel Slamka
Bc. Peter Smolinský
Bc. Ondrej Ševce
Bc. Ivan Tomovič

Tím: Agenty 007 (tím č.7)

Študijný odbor: Informačné systémy a Softvérové inžinierstvo

Akademický rok: 2008/2009

Vedúci tímu: Ing. Marián Lekavý

Obsah

1 Úvod

Tento dokument predstavuje používateľskú a systémovú príručku určenú pre používateľov Editora správania a pre prácu s agentom, ktorý vznikol v rámci Tímového projektu k téme RoboCup 3D.

2 Editor správania

Editor správania je základný projekt, ktorý bol vytvorený k téme RoboCup 3D, aby zjednodušil prácu pri vytváraní pohybov robota. Pomocou editora sa dá efektívne pracovať na ich tvorbe a následne sa dajú vytvorené pohyby aj odsimulovať v prostredí 3D futbalu.

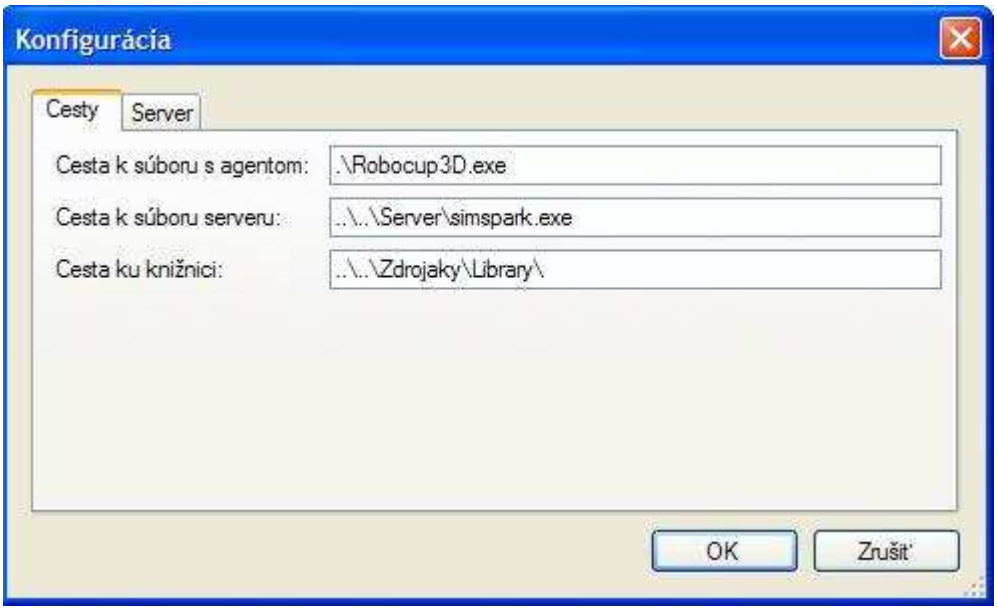
2.1. Spustenie a nastavenie editora

Samotný produkt (Editor správania) sa spúšťa ako aplikácia pomocou súboru *RobotBehaviourEditor.exe*. Na spustenie editora je potrebné mať nainštalované prostredie Microsoft .NET Framework 3.5, ktorý obsahuje balík knižníc, ktoré sú potrebné k chodu produktu.

Na to aby sa pohyby vytvorené v editore dali simulovať, je potrebné mať spustené aj ďalšie súčasti projektu. Medzi ne patrí server SimSpark, monitor a agent, ktoré budú opísané v zvlášť kapitole. Tieto súčasti treba v editorom pohybov prepojiť a to pomocou menu Možnosti -> Nastavenia (Obr. 1), kde sa k nim zadá cesta. Pričom server a monitor sa v pôvodnom stave spúšťajú automaticky naraz pomocou súboru *simspark.exe*. Po jednoduchej editácií sa dá monitor spustiť aj zvlášť pomocou súboru *monitorspark.exe*. Agent sa spúšťa pomocou súboru *Robocup3D.exe*.

Obr. 1 - Prepojenie editora s agentom a serverom

Následne sa otvorí okno Konfigurácia, v ktorom sa nastaví cesta k súboru s agentom, serverom a knižnici, ktorá obsahuje jednotlivé pohyby (Obr. 2).



Obr. 2 - Nastavenie cesty k súborom

Po tomto nastavení sa dá začať s prácou v editore a následne aj spustiť simulácia v prostredí SimSpark.

2.2. Požívateľské prostredie

2.2.1. Hlavná obrazovka

Editor správania sa skladá z hlavnej obrazovky, ktorá obsahuje základné ovládacie prvky, ktoré sú zobrazené na nasledujúcom obrázku (Obr. 3).

Obr. 3 - Hlavná obrazovka Editora správania

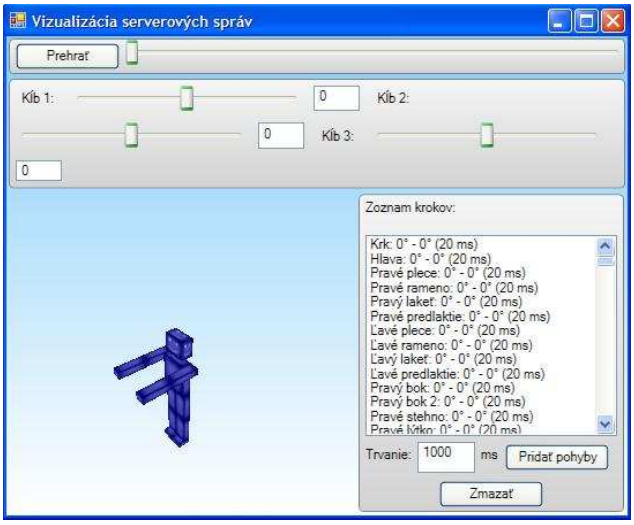
1. Hlavné menu – obsahuje položky :
    - Súbor – slúži na vytvorenie, otvorenie, uloženie a zavretie projektu
    - Správanie – slúži na pridanie a zobrazenie samotnej akcie
    - Knižnica – slúži na zobrazenie okna Knižnice prvkov
    - Možnosti – slúži na nastavenie cesty k súborom
    - Okná – slúži na prepínanie medzi jednotlivými otvorenými oknami
    - Pomocník – obsahuje informácie o programe
2. Menu okna Knižnice prvkov
    - Umožňuje pridávať jednotlivé prvky správania (pohyb, podmienku, zámer, akciu) – momentálne funkčné len pridávanie pohybu
    - Umožňuje importovať správu zo servera
3. Knižnica prvkov
    - Obsahuje jednotlivé vytvorené súbory (pohybov, zámerov, predpodmienok, akcií a správ zo servera) – momentálne funkčná knižnica pohybov a správ zo servera

2.2.2. Knižnica pohybov

Hlavnou časťou knižnice prvkov je knižnica pohybov, ktorá obsahuje jednotlivé vytvorené pohyby. Nový pohyb sa dá pridať pomocou tlačidla *Nový pohyb*, ktoré je v menu okna Knižnica prvkov. Už vytvorené prvky sa dajú editovať dvojklikom na daný pohyb.

2.2.3. Knižnica správ zo servera

Knižnica správ zo servera je súčasťou Knižnice prvkov. Do tejto knižnice sa ukladajú vykonané pohyby. Tie môžu byť následne v editore vizualizované (Obr. 4).



Obr. 4 - Vizualizácia správ zo servera

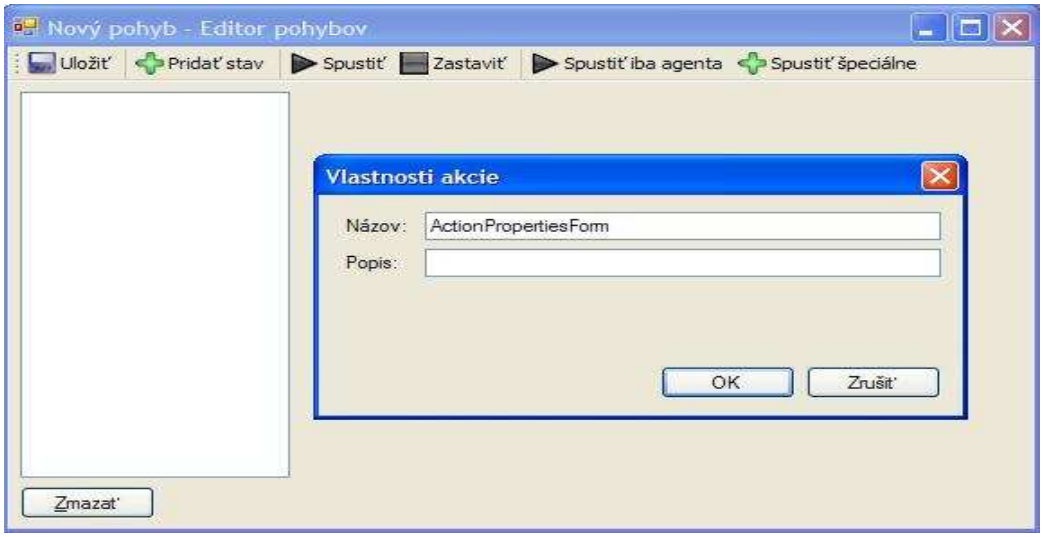
2.3. Vytvorenie a editovanie pohybu

Samotný pohyb sa skladá z akcií (stavov), ktoré môže používateľ chápať ako ucelený pohyb kĺbov (napríklad pohyb pravou nohou). Každá akcia sa následne skladá zo samotných pohybov jednotlivých kĺbov (napríklad natočenie pravého pleca o 90°). Na to aby mohol používateľ vytvárať samotné pohyby je potrebné pridať stav.

Nový pohyb sa dá vytvoriť pomocou tlačidla *Nový pohyb* v okne Knižnica prvkov. Po jeho stlačení sa otvorí okno Nový pohyb (Obr. 5).

Obr. 5 - Obrazovka okna Nový pohyb

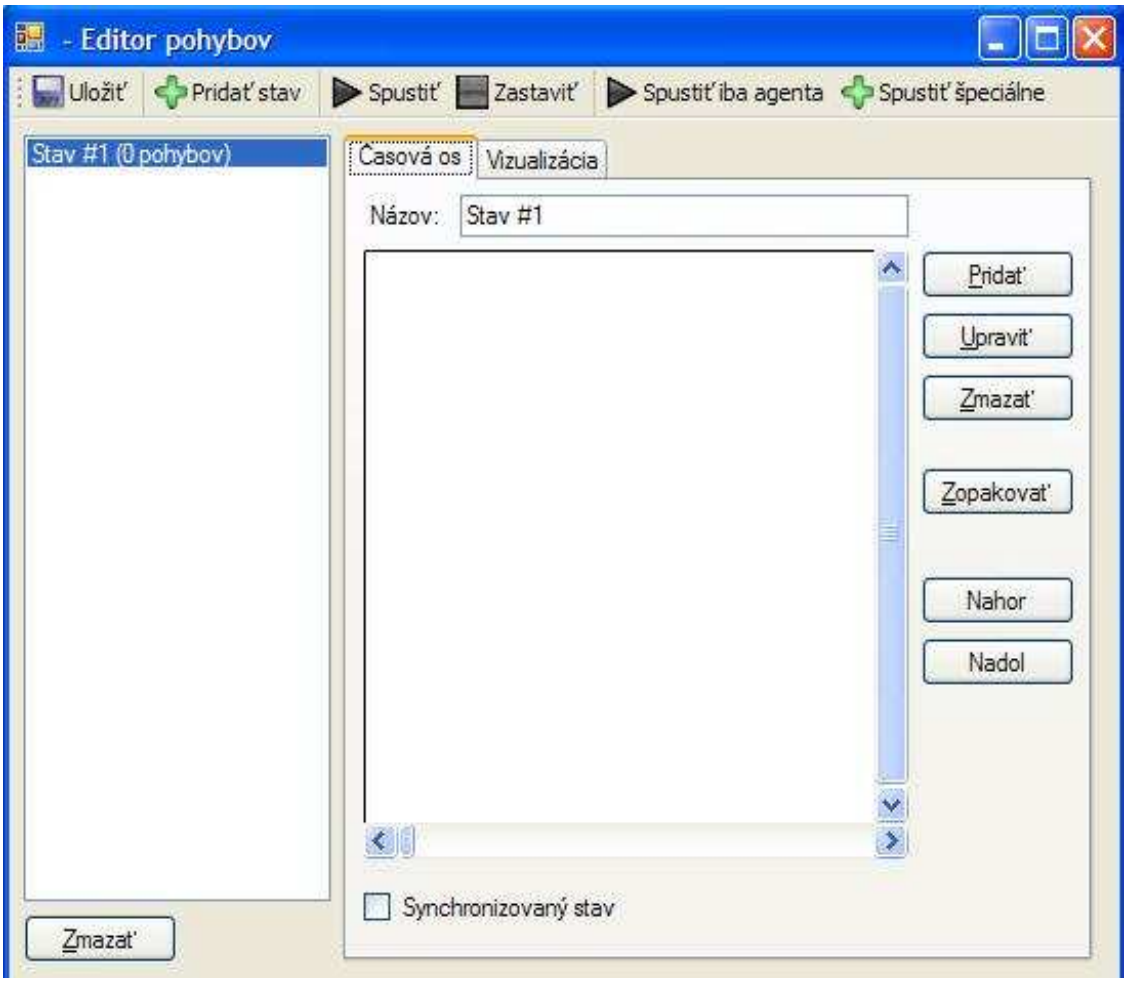
Tento nový pohyb sa dá následne uložiť pomocou tlačidla *Uložiť* z menu okna. Po jeho stlačení sa otvorí nové okno Vlastnosti akcie, ktoré obsahuje položku Názov a Popis. Položka Názov určuje, ako sa má daný pohyb volať a položka Popis, ktorá je nepovinná, určuje popis daného pohybu (Obr. 6).



Obr. 6 - Uloženie nového pohybu

Po stlačení tlačidla *Pridať stav* sa pridá v ľavom bočnom panely stav. Tento stav sa dá editovať kliknutím na jeho názov. Po danom kliknutí sa v pravom okne ukážu dve záložky, pomocou ktorých je možné vytvárať samotné pohyby.

Prvá záložka sa volá *Časová os* a ako jej názov napovedá, pôjde o editáciu pomocou pridávania jednotlivých pohybov kĺbov v časovej osi (Obr. 7).

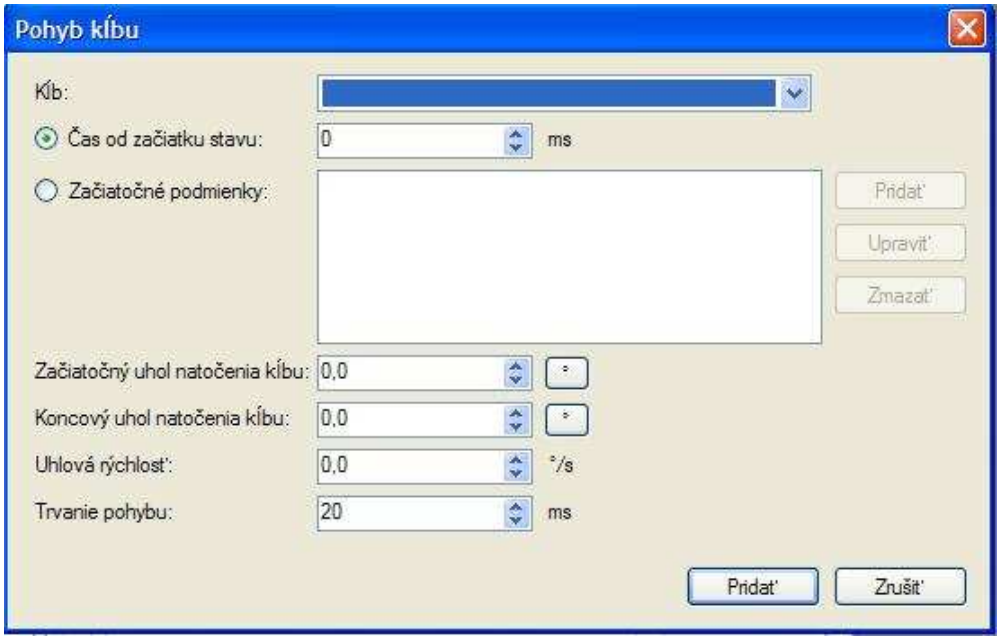


Obr. 7 - Vytvorenie pohybu pomocou časovej osi

V danom okne sa dá upraviť názov stavu a pridávať alebo editovať jednotlivé pohyby pomocou natočenia kĺbov a času trvanie daného natočenia.

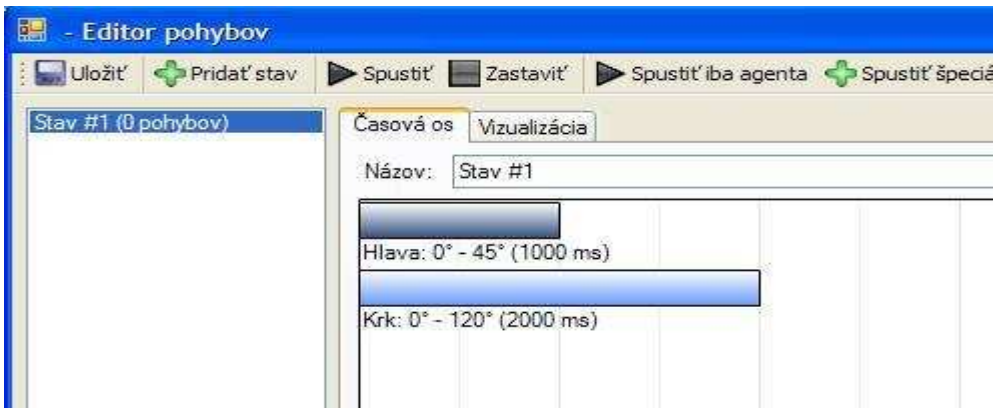
Samozrejmosťou je pridávanie viacerých stavov a pomocou nich vytvárať zložitejšie pohyby. Tieto stavy sa dajú premenovať v textovom poli Názov a taktiež zmazať pomocou tlačidla *Zmazať*, ktoré sa nachádza v spodnej časti obrazovky pod názvami jednotlivých stavov.

Ďalším krokom pri tvorbe pohybu je po vytvorení stavu pridanie samotného natočenia kĺbu a času, ktorý je potrebný na vykonanie daného natočenia. Po stlačení tlačidla *Pridať*, ktoré sa nachádza vpravo, sa otvorí nové okno Pohyb kĺbu (Obr. 8).



Obr. 8 - Nastavenie pohybu kĺbu

Toto okno obsahuje nastavenia, pomocou ktorých dokáže používateľ určiť, ktorý kĺb a ako daným kĺbom chce otáčať. Výber kĺbu, ktorý sa má otáčať, sa robí výberom z rolovacieho zoznamu všetkých kĺbov. Vedľa kĺbu sa nachádza rozmedzie uhlov, o ktoré je možné daný kĺb reálne natáčať. Po vybraní kĺbu je možné nastaviť čas, v ktorom sa má pohyb začať vykonávať a následne aj trvanie pohybu, ktoré určuje, ako dlho bude trvať natočenie kĺbu zo začiatočného stavu po koncový. Trvanie pohybu je nastavované v ms (1s = 1000ms). Následne je potrebné ešte nastaviť začiatočné a koncové uhly natočenia daného kĺbu. Po týchto nastaveniach sa dá dané natočenie uložiť pomocou tlačidla *Pridať*. To spôsobí, že sa dané natočenie kĺbu zobrazí v záložke Časová os (Obr. 9).

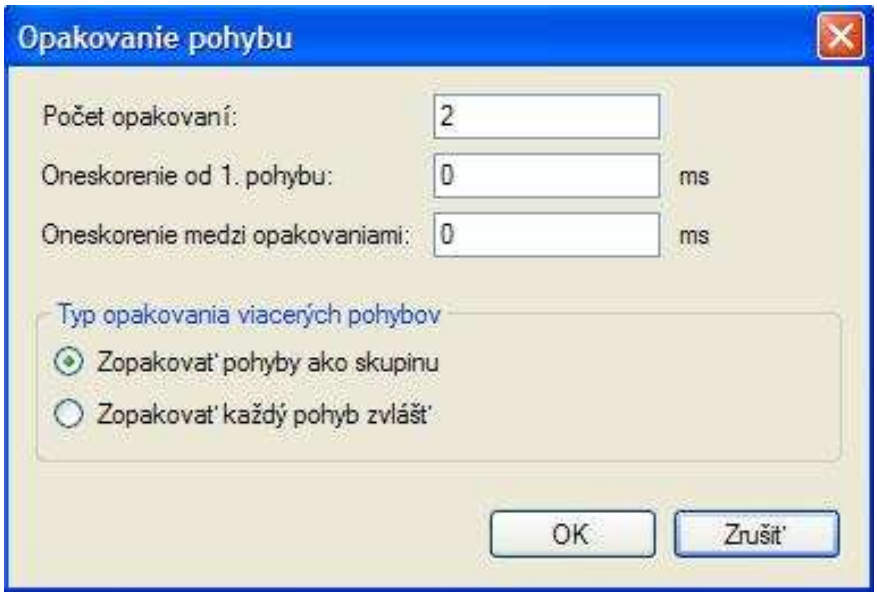


Obr. 9 - Pridanie natočenia kĺbov

Na záložke Časová os je zobrazenie jednotlivých natočení, ktoré používateľ pridal. Tieto zobrazenia sú farebne rozlíšené, čo zvyšuje prehľadnosť jednotlivých pohybov kĺbov. Každé takéto natočenie, ktoré je zobrazené, obsahuje názov kĺbu, natočenie v nastavených uhloch začiatku a konca a čas, ktorý má trvať dané natočenie.

Jednotlivé natočenia sa dajú následne editovať pomocou tlačidla *Upraviť* alebo vymazať pomocou tlačidla *Zmazať*. Takto sa dá pracovať s jednotlivými natočeniami zvlášť alebo je možné pomocou klávesy CTRL označiť aj viaceré natočenia, ktoré je možné posúvať vľavo a vpravo, skracovať alebo predlžovať na časovej osi a tým určiť ich vykonávanie vzhľadom na čas. Jednotlivé natočenia kĺbov je možné pre zlepšenie prehľadnosti zoraďovať od hora dole pomocou tlačidiel *Nahor* a *Nadol*.

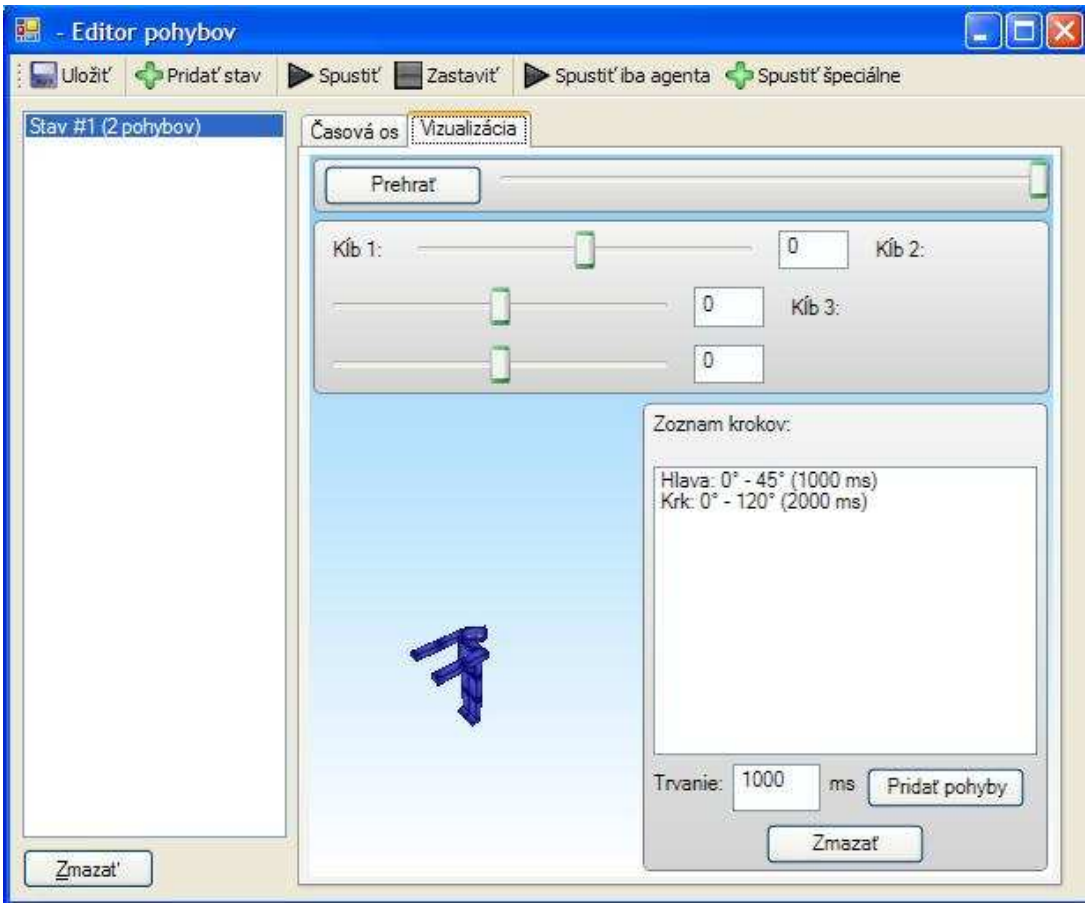
Po stlačení tlačidla *Zopakovať* sa otvorí nové okno Opakovanie pohybu (Obr. 10). V tomto okne sa dá následne nastaviť zopakovanie vybraných natočení kĺbov. Netreba preto manuálne vkladať rovnaký pohyb, ale stačí len nastaviť počet jeho opakovaní a oneskorenia po jednotlivých vykonávaniach.



Obr. 10 - Opakovanie pohybu

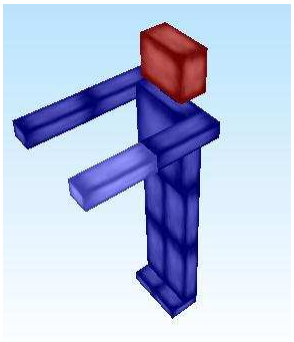
Následne sa takto vytvorený pohyb, ktorý sa skladá z viacerých natočení kĺbov, dá pomocou druhej záložky prehrať.

Druhá záložka sa volá Vizualizácia (Obr. 11). V tejto záložke je možné prehrať už vytvorený pohyb pomocou tlačidla *Prehrať*, ktorý používateľ vytvoril v záložke Časová os. Taktiež je však možné pomocou modelu robota vytvárať pohyb jednotlivých kĺbov. Takéto vytváranie pohybov môže byť pre používateľa jednoduchšie, nakoľko si vie ľahšie predstaviť, čo daný robot vykoná natočením určeného kĺbu.



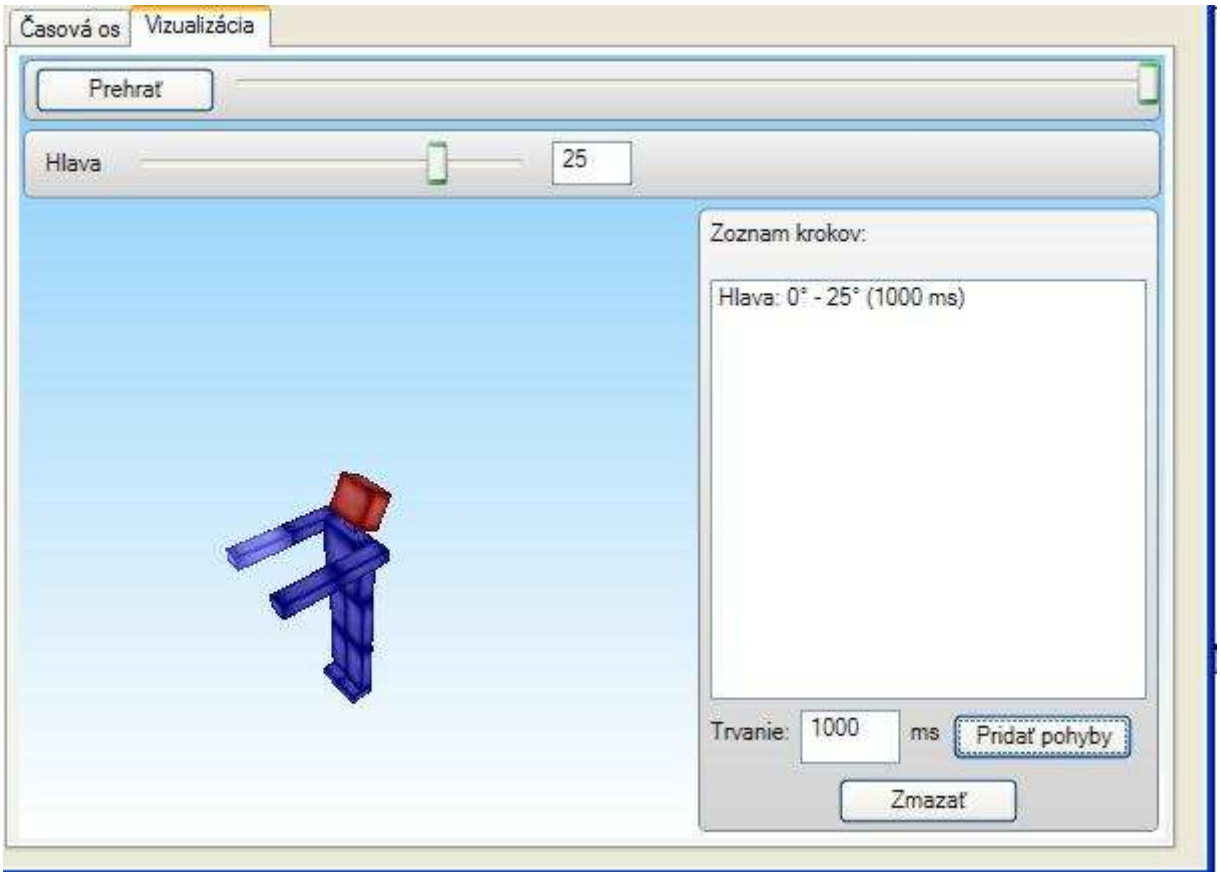
Obr. 11 - Vizualizácia v editore

K tomu, aby mohol používateľ vytvárať pomocou tohto modelu pohyby, je potrebné vybrať kĺb, ktorým chce pohnúť o požadovaný uhol. Tento výber sa robí kliknutím na daný kĺb, ktorý sa následne sfarbí červenou farbou (Obr. 12).



Obr. 12 - Nastavenie editovaného kĺbu

Následne sa pomocou posuvného bežca nastaví uhol, o ktorý sa má daný kĺb ohnúť. Tento uhol sa dá nastaviť aj vo vedľajšom okne vpísaním daného uhla. Potom je potrebné určiť trvanie pohybu v ms (1s = 1000ms) a stlačí sa tlačidlo *Pridať pohyby* (Obr. 13).



Obr. 13 - Pridanie pohybu pomocou vizuálizácie

Takýmto spôsobom sa dá pridať natočenie jedného, ale aj viacerých kĺbov. Po stlačení tlačidla *Pridať pohyby* sa zobrazí v pravom panely zoznam krokov všetkých kĺbov, ktoré sa pridali. Tento zoznam obsahuje kĺb, jeho natočenie a dĺžku trvania otáčania kĺbu. Jednotlivý kĺb sa dá z tohto zoznamu vymazať tlačidlom *Zmazať*.

Takto vytvorené natočenia kĺbov sa pomocou tlačidla *Prehrať* môžu prehrať na modely, ktorý zobrazí simuláciu daných natočení. Táto simulácia je bez reálnej fyziky robota, ktorá sa nachádza na servery a slúži len na ukážku a porovnania činnosti, ktorú chceme aby agent vykonal.

Posledným krokom pri vytváraní alebo editácií pohybov je uloženie zmien, ktoré sa vytvorili. Na to, aby sa tieto zmeny uložili do súboru, ktorý vie agent načítať, je treba stlačiť tlačidlo *Uložiť*, ktoré sa nachádza v menu nad menami stavov. Po tomto uložení je možné daný súbor poslať hráčovi na spracovanie.

2.4. Vykonanie pohybu

Keď si vytvoríme alebo zeditujeme pohyb, je potrebné otestovať, či je tento pohyb naozaj použiteľný v reálnom prostredí, na ktorom beží RoboCup 3D. V knižnici pohybov sú už vytvorené základné pohyby končatinami, vsávanie, chôdza, ktoré slúžia ako ukážka pre budúcich používateľov. Tieto pohyby sú uložené v súboroch s názvami **.rmo*.

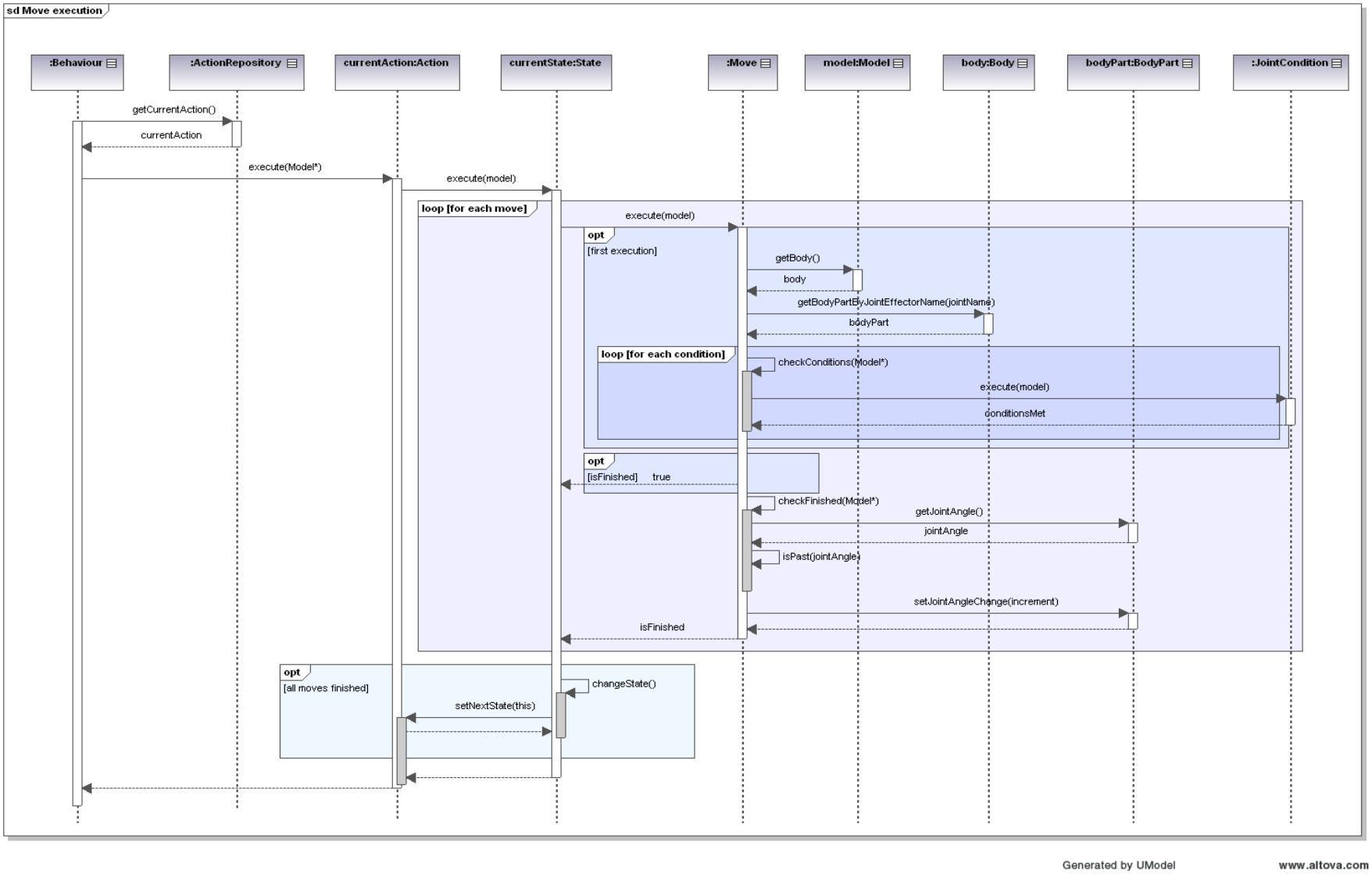
Editor správania poskytuje možnosť takýto súbor priamo poslať hráčovi na vykonanie. Je však nutné mať správne nastavené cesty k jednotlivým súborom servera, agenta a knižnici s jednotlivými položkami (Obr. 2) podľa návodu, ktorý je v kapitole 2.1.

Ak máme všetko správne nastavené, je možné poslať súbor agentovi. To sa spraví jedným stlačením tlačidla *Spustiť*. Toto stlačenie spustí server spolu s monitorom a následne do neho pripojí hráča, ktorý prečíta, čo má vykonať a následne v reálnom prostredí vykonáva vytvorené pohyby. Postup vykonávania je zobrazený na sekvenčnom diagrame na nasledujúcej strane (Obr. 15).

V prípade, že máme už spustený server a chceme spustiť iba hráča tak stačí ak stlačíme tlačidlo *Spustiť iba agenta*. V prípade potreby môžeme agentovi nastaviť aj špeciálne spustenie pomocou prislúchajúceho tlačidla *Spustiť špeciálne*. Po jeho stlačení sa zobrazí okno Nastavenie spustenia (Obr. 14), v ktorom sa dá nastaviť vykonávanie. Je možné ho spomaliť alebo zrýchliť, nastaviť oneskorenie začiatku vykonávania a taktiež nastaviť prestávky medzi jednotlivými stavmi.

Obr. 14 - Okno slúžiace na nastavenie špeciálneho spustenia

Po stlačení tlačidla *Spustiť* sa následne spustí agent a vykoná zvolené pohyby upravené podľa daného špeciálneho spustenia.



Obr. 15 - Sekvenčný diagram vykonania pohybu

2.5. Message Parser

Pre lepšie pochopenie práce servera sme vytvorili mechanizmus, ktorým je možné ľubovoľný pohyb robota, vykonaný na servery SimSpark, zaznamenať, preniesť do editora pohybov a tam ho prehrať. Takto zaznamenané pohyby sa nachádzajú v Knižnici správ zo servera, ktorá sa spomína v kapitole 2.2.3.

2.5.1. Implementácia spätnej väzby (Server -> Editor)

Následne bolo potrebné vytvoriť nasledovnú funkcionalitu:

- Zaznamenávanie správ zo servera, ktoré prijme agent,
- Parsovanie správ na strane editora správania a vloženie rozpoznaných pohybov do pripravených dátových štruktúr,
- Vykonanie a vykreslenie pohybu uloženého v daných štruktúrach.

2.5.2. Zaznamenávanie správ agentom

Agentovi sme implementovali možnosť prijímať argumenty z príkazového riadku. Pre účely zaznamenávania serverových správ bol zavedený argument –message, ktorý prijíma ako parameter názov súboru, do ktorého budú logované správy. V prípade, že je daný argument vynechaný, logovanie serverových správ bude vypnuté. Treba zdôrazniť, že ide o úplne samostatné logovanie, ktoré je nezávislé od debugovacích, chybových a iných výpisov.

2.5.3. Parsovanie správ v editore

Editor obsahuje triedu ServerMessageParser, v ktorej je uložená metóda ParseMotions(), vracajúca zoznam pohybov zaznamenaných v súbore. V súbore sú zaznamenané aj statické polohy kĺbov, aj keď s nimi nebolo v danej iterácií pohybované. Takéto záznamy sú rozpoznané a nevytvárajú sa na základe nich pohyby, keďže reálne žiadny pohyb nereprezentujú. Medzi pohyby sú vložené iba záznamy, ktoré reálne afektujú pohyb hráča. Pri parsovaní súboru sú načítavané aj hodnoty gyroskopu.

2.5.4. Vykonanie pohybu v editore

Na vykonanie zaznamenaného pohybu v editore je použitý engine, ktorý sa používa aj na vykonávanie modelovaných pohybov. Toto je zabezpečené použitím rovnakého rozhrania (dátovej štruktúry JointMotion) pri modelovaní nových pohybov, aj pri prehrávaní zaznamenaných pohybov.

3 Server SimSpark

Na začiatku práce na RoboCup 3D sme dostali od predchádzajúceho tímu Hviezdna 11 skompilovaný server SimSpark. Tým pádom sme nemuseli už pracne kompilovať server, ale mohli sme sa zamerať na iné činnosti spojené s prácou na projekte RoboCup 3D.

3.1. Vypnutie monitoru

Skompilovaný server sa spustí pomocou súboru *simspark.exe* a automaticky spúšťa monitor, ktorý slúži na vizualizáciu. Kvôli zníženiu nárokov na počítač sa dá tento monitor spustiť aj samostatne a nastaviť mu nižšia prioritu vykonávania.

Pre spustenie servera SimSpark bez monitora je potrebné v súbore *simspark.rb* nájsť položku $enableInternalMonitor a nastaviť ju na hodnotu false. Toto nastavenie spôsobí, že po spustení servera (*simspark.exe*) sa už monitor automaticky nespustí. Následne je však nutné pre vizualizáciu spustiť monitor manuálne cez súbor *monitorspark.exe*. A cez task manager znížiť prioritu danému procesu.

Pre opätovné zapnutie automatického spúšťania servera spolu s monitorom treba znovu zmeniť hodnotu položky $enableInternalMonitor na hodnotu true.

3.2. Pridanie viacerých hráčov

V prípade potreby pridania viacerých hráčov je potrebné použiť prepínač –pid. Tento prepínač priradí agentovi číslo v tíme a tak sa umožní spustenie viacerých agentov naraz.

Ak potrebujeme priradiť agentov do rôznych tímov je možné použiť prepínač –tname. Ten nastaví agentovi príslušnosť k tímu. Základná hodnota, keď sa prepínač nepoužije, je Agenty007.

Počas jedného spustenia servera sa môžu použiť iba dva názvy tímov, jeden náš a jeden oponentov. Ak sa použije aj tretí názov tímu, server pridá hráča bez príslušnosti k tímu do stredu ihriska.

3.3. Pripojenie na server na inom PC

Agent sa automaticky pripojí na localhost (127.0.0.1). V prípade potreby pripojiť agenta na server, ktorý beží na inom počítači, je potrebné použiť prepínač –ip a zadať IP adresu cieľového počítača so serverom (napríklad -ip 192.168.5.3). Treba mať vo firewalle povolený TCP port 3100. Po tomto nastavení je možné prepojiť server a agenta na rôznych počítačoch.

4 Agent Robocup3D

Agent bol vyvíjaný v prostredí Microsoft Visual Studio 2008. Pre beh agenta bez Visual Studia je postačuje mať nahrané nasledujúce knižnice: msvcm90d.dll, msvcp90d.dll a msvcr90d.dll.

4.1. Vykonávanie pohybu

Samotné vykonávanie pohybu sa uskutočňuje načítaním a následným rozparovaním údajov. Tieto údaje sa uložia do pripravených štruktúr a sú pripravené na následné vykonanie agentom. Toto vykonanie je zobrazené pomocou sekvenčného diagramu (Obr. 15).

4.2. Logovanie v agentovi

Na to, aby sa dali ľahšie odhaľovať vzniknuté chyby, sme vytvorili možnosť logovať správanie agenta. Samotné logovanie sa rozdeľuje na dve základné skupiny a to na logovanie chýb, ktoré vzniknú počas behu programu a logovanie pomocných debug výpisov, ktoré umožňujú efektívnejšie zisťovať správanie sa agenta.

4.2.1. Parametre príkazového riadku

Nastavenie logovania je možné spustiť cez príkazový riadok pomocou zadania nasledujúcich prepínačov:

- -debug "path"
    - tento prepínač spôsobí ukladanie debug výpisov do súboru nachádzajúceho sa v ceste "path". Ak súbor existuje bude výpis pripojený na jeho koniec a ak by súbor neexistoval bude automaticky vytvorený.
    - v prípade, že nie je uvedený parameter "path" pri spustení, debug výpisy sa zapíšu do súboru s názvom logXXXXXXXXXX.txt v adresári, v ktorom sa nachádza agent, pričom XXXXXXXXXX je časová známka slúžiaca na to, aby sa výpisy z rôznych spustení nachádzali v rozličných súboroch.
- -error "path"
    - spôsobí ukladanie záznamu chyb do súboru nachádzajúceho sa v ceste "path". Ak daný súbor existuje bude výpis pripojený na jeho koniec a ak by súbor neexistoval bude automaticky vytvorený.
    - ak nie je uvedený parameter "path", záznamy sa budú ukladať vždy na koniec súboru s názvom log.txt.

- -nodebug
    - tento prepínač zabráni výpisu a ukladaniu debug výpisov, pričom prekonáva nastavenia prepínača -debug

4.2.2. Ukážka použitia logovania priamo v kóde agenta

Pri chybe počas behu programu je potrebne ošetriť vzniknutú chybu. Toto ošetrenie je podobné ako v nasledujúcom príklade:

```
try
{
        //volania, ktoré môžu spôsobiť chybu alebo výnimku
}
catch(string &err)
{
        Logger::logError(err);
        //ďalší error handling, podľa potreby
}
```

V prípade potreby debug výpisov na obrazovku a do súboru sú potrebne nasledovné nastavenia. Je potrebné odvodiť klientskú triedu, ktorá chce ukladať do logu informácie od rozhrania Loggable. Následne treba implementovať rozhranie, teda public metódy:

- string getStateAsString(Model *model)
    - táto metóda vráti údaje, ktoré chceme zapísať do logu ako string, pričom je vhodné použiť preťažené metódy Logger::concatenate(), ktoré naformátujú reťazec
- string getInvokerName()
    - táto metóda vráti názov triedy

Na koniec je potrebné implementovať metódu, ktorá vykoná zápis pomocou metódy Logger::write(Loggable* obj, Model* model). Detaily k implementácií sú viď triedy Move a State.

4.3. Pridanie nového prepínača

Parsovanie argumentov funkcie main sa vykonáva vo funkcii parseArgs(int argc, char* argv[]) v súbore *main.cpp* a vykonáva sa pomocou jednoduchých 'if'-ov a porovnávania reťazcov.

Pridanie nového parametra je potrebné vykonať nasledujúce kroky:

- pridanie novej položky do enumerácie Option v súbore *ConfigRepository.h*
- vyparsovanie nastavení z príkazového riadku vo funkcii parseArgs
- vyvolanie inštancie repozitára zavolaním "ConfigRepository cr = ConfigRepository::instance()"
- uloženie nastavení do repozitára volaním "cr->add(Option::MojaNovaPolozka,mojRetazecSNastavenim);
- použitie nastavenia v klientskej triede zavolaním "cr->contains(Option::MojaNovaPolozka)" alebo "string nastavenie = cr->get(Option::MojaNovaPolozka)"

4.4. Rovnovážny modul v RoboCup 3D

Robot NAO je humanoidný robot so 22 pohyblivými kĺbmi. Jeho pohyb je komplexný problém. Pohyby kĺbov menia polohy rôznych častí robota, ktoré zväčša obsahujú ďalšie kĺby. Robot pri pohybe relatívne ľahko stratí rovnováhu. Preto časť nášho úsilia bola smerovaná k analýze fyziky robota a jeho následnej stabilizácii. Náš postup môžeme rozdeliť do viacerých krokov

4.4.1. Vypočítanie aktuálnej pozície častí tela robota

Najskôr sme vyhľadali a zosumarizovali všetky informácie o začiatočnej pozícii a konfigurácii robota. Jedná sa o údaje ohľadom hmotností, rozmerov, počiatočnom rozmiestnení, tvarov častí tela, pozícii a osí otáčania kĺbov v rámci nich. Údaje sme čerpali hlavne z oficiálneho manuálu a zdrojových kódov servera. Sú spísane v tabuľke na obrázku nižšie (Obr. 16). Z daných údajov vieme zostaviť model v začiatočnej pozícii hráča po jeho pripojení na server. Nasledujúce pozície sú ovplyvnené rotáciami kĺbov a celkovou polohou hráča v rámci ihriska (hlavne rotácia vzhľadom na zem). Z týchto údajov dokážeme vypočítať aktuálne pozície jednotlivých častí tela (vzhľadom na zem hracej plochy). Výpočet zahrňuje hlavne rotácie bodov okolo vektora (os otáčania kĺbu). Pre dané výpočty požívame Rodrigézovu metódu.

Name	Parent	Translation	Mass	Geometry	Name	Anchor	Axis	Min	Max
neck	torso	0, 0, 0.09	0.05	Cylinder L: 0.08 R: 0.015	HJ1	0, 0, 0	0,0,1	-120	120
head	neck	0, 0, 0.065	0.35	Sphere 0.065	HJ2	0, 0,-0.005	1,0,0	-45	45
shoulder	torso	0.098, 0, 0.075(r) -0.098, 0, 0.075(l)	0.07	Sphere 0.01	AJ1	0, 0, 0	1,0,0	-120	120
upperarm	shoulder	0.01, 0.02, 0(r) -0.01, 0.02, 0(l)	0.150	Box 0.07, 0.08, 0.06	AJ2	-Translation	0,0,1	-95(r) -1(l)	1(r) 95(l)
elbow	upperarm	-0.01, 0.07, 0.009(r) 0.01, 0.07, 0.009(l)	0.035	Sphere 0.01	AJ3	0, 0, 0	0,1,0	-120	120
lowerarm	elbow	0, 0.05, 0	0.2	Box 0.05, 0.11, 0.05	AJ4	-Translation	0,0,1	-1(r) -90(l)	90(r) 1(l)
hip1	torso	0.055, -0.01,-0.115(r) -0.055, -0.01,-0.115(l)	0.09	Sphere 0.01	LJ1	0, 0, 0	-0.7071,0,0.7071 (r) -0.7071,0,-0.7071(l)	-90	1
hip2	hip1	0, 0, 0	0.125	Sphere 0.01	LJ2	0, 0, 0	0,1,0	-45(r) -25(l)	25(r) 45(l)
thigh	hip2	0, 0.01, -0.04	0.275	Box 0.07, 0.07, 0.14	LJ3	-Translation	1,0,0	-25	100
shank	thigh	0,0.005,-0.125	0.225	Box 0.08, 0.07, 0.11	LJ4	0,-0.01, 0.045	1,0,0	-130	1
ankle	shank	0, -0.01,-0.055	0.125	Sphere 0.01	LJ5	0, 0, 0	1,0,0	-45	75
foot	ankle	0, 0.03,-0.035	0.2	Box 0.08, 0.16, 0.03	LJ6	0,-0.03, 0.035	0,1,0	-25(r) -45(l)	45(r) 25(l)
Torso			1.2171	Box 0.1, 0.1, 0.18					

Obr. 16 - Konfigurácia robota Nao

Vysvetlivky:

- Name – meno časti tela
- Parent – rodičovská časť tela, na ktorú je daná časť naviazaná
- Translation – posunutie bodu pre danú časť tela od rodičovskej
- Mass – hmotnosť príslušnej časti tela
- Geometry – geometria, grafický vzhľad časti tela, pre fyzikálny model hráča nie je podstatná
- Name – meno kĺbu v danej časti tela
- Anchor – miesto umiestnenia kĺbu vrámci časti tela (posunutia od rodičovskej časti)
- Axis – os otáčania kĺbu
- Min – minimálny uhol natočenia
- Max – maximálny uhol natočenia

4.4.2. Identifikovanie ťažiska robota

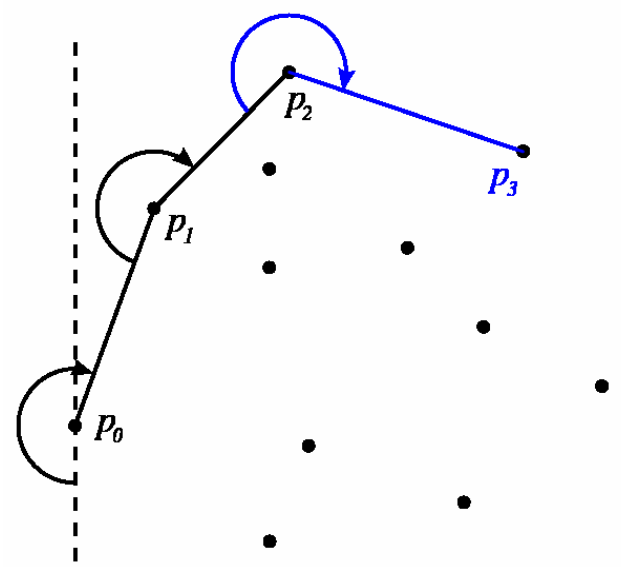
Ide o výpočet, ktorý podľa aktuálnych pozícii časti tela robota a ich hmotností určí ich spoločné ťažisko. Daný bod máme aj za pomoci perceptora gyroskopu v súradniciach relatívnych vzhľadom na zem ihriska.

4.4.3. Identifikovanie oporných bodov robota

Vychádzajúc so súradníc bodov častí tela, ktoré sú relatívne vzhľadom na zem ihriska, sme vybrali tie najnižšie dotýkajúce sa zeme. Jedná sa o oporné body. Tieto slúžia ako vstup ďalším výpočtom.

4.4.4. Vypočítanie stabilnej plochy robota

Ide o plochu, ktorá ohraničuje stabilnú polohu hráča. Ťažisko hráča sa musí nachádzať nad touto plochou, inak mu hrozí preváženie do niektorej zo strán a pád. Daná podmienka musí platiť pri tzv. statických pohyboch (kebyže hráč zastane v ktoromkoľvek momente, nespadol by). Statické pohyby plne postačujú robotovi pre hru futbal. Pri dynamických pohyboch by bolo rovnako potrebné uvažovať ťažisko a plochu stability, akurát by sa ťažisko mohlo nachádzať aj mimo nej v smere pohybu robota. Plochu stability počítame ako konvexný obal oporných bodov hráča. Na výpočet konvexného obalu sme použili algoritmus balenia darčeka nazývaný tiež Jarvis march (podľa autora R. A. Jarvisa). Základnou myšlienkou algoritmu je postupne obaľovanie množiny bodov z vonkajšej strany (Obr. 17).



Obr. 17 - Gift Wrap Algoritmus

4.4.5. Určenie vektoru posunu hráča do stabilnej polohy

Pomocou priemetu ťažiska do roviny konvexného obalu a výpočtu stredu konvexného obalu vieme určiť vektor posunutia hráča do stabilnej polohy. Keď sa hráčovo ťažisko dostane na hranice stabilnej plochy, vypočítame vektor jeho posunutia tak, aby sa zabezpečila stabilizácia hráča.

5 Inštalácia produktu

Táto kapitola obsahuje postup inštalácie jednotlivých častí projektu RoboCup, ktoré sú potrebné na spustenie simulácie zápasu.

5.1. Minimálne požiadavky

Na spustenie a sledovanie simulácie je potrebný operačný systém Windows XP/2000/Vista alebo operačný systém Linux/UNIX. Hardvérové nároky nie sú veľké a preto postačuje konfigurácia, ktorú odporúča výrobca použitého operačného systému.

Pre spustenie servera je dobré mať silnejší aspoň dvojjadrový procesor, ktorý umožní lepšie spracovávanie jednotlivých procesov (server a monitor).

5.2. Inštalácia servera

Spolu s produktom je dodaný aj server SimSpark, ktorý zabezpečuje simuláciu zápasu. Server SimSpark je skompilovaný a pre operačný systém Windows sa nachádza v súbore *Server_SimSpark.rar*. Tento súbor stačí rozbaliť a server SimSpark je pripravený na spustenie pomocou súboru *simspark.exe*. Na spustenie simulácie je potrebné nastaviť firewall operačného systému tak, aby umožňoval pripojenie hráčov k serveru. V operačnom systéme Windows XP, ak je zapnutý firewall, tak stačí po spustení serveru kliknúť na možnosť odblokovať.

5.3. Inštalácia monitora

Spolu so serverom SimSpark je dodávaný aj monitor, ktorý slúži na sledovanie zápasu. Monitor je taktiež skompilovaný a dá sa spustiť pomocou súboru *monitorspark.exe*, ktorý sa nachádza taktiež v súbore *Server_SimSpark.rar*.

5.4. Inštalácia agenta a editoru pohybov

Náš výsledný produkt Editor správania a agent Robocup3D je v súbore Build_v.0.67.rar. Tento súbor obsahuje poslednú verziu buildu editora a agenta. Súčastne sa v ňom nachádza aj adresár Library, v ktorom sú uložené vytvorené pohyby. Na to aby išiel hráč spustiť aj bez prostredia Microsoft Visual Studio sú v danom adresári pribalené aj potrebné knižnice pre spustenie. Pred samotným spustením hráča je potrebné nastaviť aktuálne cesty viď kap. 2.1.