

ROBOCUP – tretí rozmer

(dokumentácia k 2. šprintu)

Tím č.17 : Tím 17 žije...

Bc. Filip Baďura
Bc. Roman Bilevic
Bc. Tomáš Blaho
Bc. Andrej Bisták
Bc. Peter Holák
Bc. Jozef Macho
Bc. Peter Paššák

1. Analýza

1.1. Jednoduché pohyby

Táto podkapitola sa zameriava na analýzu jednoduchých pohybov, ktoré implementoval tím Androids. V prvej časti sú uvedené pohyby implementované do prototypu a v druhej časti pohyby, ktoré obsahuje finálny produkt. Informácie sa čerpané z [1] – dokumentácie tímu Android.

1.1.1. Základné pohyby implementované v prototypu

➤ Vstávanie

Androids vytvorili 2 typy vstávania a to vstávanie z chrbta a vstávanie z brucha, pričom ako základ využili vstávanie JIM-a.

Implementovali aj pomocné pohyby pád vpred a vzad, ktoré používali pri testovaní vstávania.

• *Vstávanie z brucha*

Odstránili dlhé čakacie fázy a niekoľko zbytočných pohybov kĺbov, ktoré už pred vykonaním zmeny boli v koncovej polohe vďaka niektorej skoršej fáze pohybu. Upravili záver pohybu tak, aby hráč po

jeho vykonaní ostal stáť v základnom postoji, do ktorého ho nastaví server po nabeamovaní, keďže pôvodný pohyb končil pádom hráča späť na zem. Pohyb prebieha v nasledovných krokoch:

- najskôr hráč uvedie všetky kĺby do neutrálnej polohy, okrem ramenných, ktoré pripaží
- hráč rozkročí a úplne skrčí nohy
- postupne prinožuje a vystiera nohy, čím zároveň vstáva
- v koncovej fáze vystierania hráč predpaží ruky, ktoré až do tejto fázy boli pripažené

• *Vstávanie z chrbta*

U tohto pohybu vykonali rovnaké úpravy ako u vstávania z brucha a aj priebeh pohybu je takmer totožný s výnimkou začiatku.

Fázy pohybu sú:

- vystretie tela a pripaženie rúk
- rýchle predpaženie a čiastočné rozkročenie, vďaka čomu sa hráč čiastočne posadí

- dokončenie rozkročovania a pokrčenia nôh, a zároveň opätovné pripaženie, čím sa hráč prevráti na brucho
- hráč rozkročí a úplne skrčí nohy
- postupne prinožuje a vystiera nohy, čím zároveň vstáva
- v koncovej fáze vystierania hráč predpaží ruky, ktoré až do tejto fázy boli pripažené

Aj toto vstávanie úspešne otestovali na 100%.

➤ **Pád vpred a vzad**

Tieto pomocné pohyby sú založené na jednoduchej rýchlej zmene uhla kĺbu členku, ktorý je zodpovedný napnutie a pritiahnutie nártu. Boli vyvinuté čisto pre potreby testovania vstávania.

➤ **Chôdza**

Implementovali veľmi drobnú, pomalú chôdzu dopredu a dozadu, pričom obe chôdze vznikli úpravou prototypu chôdze, ktorý prevzali spolu s agentom JIM. Vytvorili aj verziu rýchlej chôdze dopredu, ktorá je založená na drobčení.

• ***Drobná pomalá chôdza dopredu***

Oproti zdedenej chôdzi je tento typ chôdze vzpriamenejší a podstatne drobnejší. Dĺžka kroku je približne pätina dĺžky hráčovho chodidla. Táto chôdza je veľmi pomalá a jej účelom je predovšetkým presné priblíženie k lopte bez jej odkopnutia. Pri tejto aj ostatných implementovaných chôdzach hráč výrazne pracuje s rukami, keďže ruky sú dôležitým prvkom v udržiavaní rovnováhy a prenášaní ťažiska.

Periebeh pohybu:

- na začiatku kráčania robot pokrčí ruky aj nohy, aby zvýšil svoju stabilitu a jeden krok následne pozostáva z dvoch fáz
- hráč preniesie váhu na opornú nohu a druhú nohu zdvihne a mierne vystrie v kolene, čím ju predsunie pred opornú nohu, zároveň rukou na strane opornej nohy pohne v ramene dopredu
- dostúpi na predsunutú nohu a preniesie na ňu váhu kvôli ďalšiemu kroku

Testovaním zistili, že chôdza je stabilná a hráč nespadol. Jeden krok trvá 300 ms, inicializačná fáza 700 ms. Na 90% bola táto chôdza pri testovaní priamočiara.

• ***Drobná pomalá chôdza dozadu***

Práca rúk je rovnaká ako v predošlom opísanom spôsobe chôdze. Dĺžka kroku je približne tretina chodidla

Tento pohyb pozostáva z nasledovných úkonov:

- mierne odrazenie sa z päty pri predkopnutí nohy, lebo päta sa jemne zachytí o zem
- mierne zanoženie pri vrátení predkopnutej nohy späť na zem

Táto chôdza bola pri testovaní taktiež stabilná. Krok trvá 300 ms, inicializácia 700 ms, výraznejšie však mení smer.

- ***Drobčivá rýchla chôdza dopredu***

Podobá sa drobnej chôdzi dozadu, hráč pri nej mierne nadskakuje. Rozdielom voči pohybu JIM-a je menší predklon hráča, vďaka čomu sa nestáva, že by padol na nos. Táto chôdza má rovnaké fázy ako predošlé dve chôdze, odlišuje sa len v hodnotách otočení jednotlivých kĺbov a je pri nej výraznejšie prenášanie váhy na opornú nohu pri kroku. Je určená hlavne na prekonávanie väčších vzdialeností

- **Otáčanie**

Implementovali otočenie o 90° do oboch smerov.

- ***Otočenie o 90°***

Toto otočenie okrem fázy inicializácie pracuje len s kĺbmi nôh. Je implementované do oboch smerov a založené na humanoidnom spôsobe otáčania. Priebeh jedného cyklu otočenia je nasledovný:

- mierne pokrčenie nôh a rúk v inicializačnej fáze pre zvýšenie stability priblížením ťažiska k zemi
- prenesenie váhy na opornú nohu a otočenie druhej nohy v bedrovom kĺbe úplne na stranu
- prenesenie váhy na otočenú nohu a prinoženie pôvodnej opornej nohy
- uvedenie pohybovaných kĺbov do pozície, v ktorej boli na konci inicializačnej fázy pred ďalším pokračovaním

Pri úspešne vykonaných cykloch sa hráč takmer nepohne z miesta, čo je výrazné zlepšenie v porovnaní so zdedeným otáčaním. Problémom je nedeterminizmus cyklu, keďže kvôli nedoladenému prenosu váhy v počiatočných fázach hráč približne v 50% vykonaných cyklov nezmení svoje natočenie. Ďalším nedostatkom je fakt, že ani pri úspešnom vykonaní cyklu nie je výsledný uhol otočenia 90°, ale len približne 70°. Toto je pravdepodobne tiež spôsobené chybou v prenose váhy počas cyklu.

- **Kop do lopty**

Naimplementovali dva typy kopu:

- priamy kop do lopty

- kop do lopty hranou chodidla

- ***Priamy kop do lopty***

Ide o kop špicou chodidla pri zapojení bedrového, kolenného kĺbu a členku. Ide o kop, ktorý slúži na ďaleké prihrávky a kopy na bránu.

Pohyb tvorí päť fáz:

- počiatočná fáza prikrčenia sa dostane robota do polohy, z ktorej môže kop vykonať čonajstabilnejšie. Prikrčenie posunie ťažisko robota nižšie a zvýši tak stability a do tejto fázy sa zapájajú kĺby bedier, kolien a členkov
- druhá fáza vychýlenia tela do strany získa pre agenta priestor na samotný kop a zabezpečí, že po zdvihnutí nohy (opačnej ako je smer naklonenia agenta) sa agent neprevráži na opačnú stranu. V tejto časti pohybu sa okrem kĺbov členkov a bedier zapájajú aj ramenné kĺby ruky
- treťou fázou je zodvihnutie nohy. Ide o spoluprácu bedrového, kolenného kĺbu a kĺbu členka s cieľom nezavadiť nohou počas zdvíhania o podložku
- štvrtá fáza je samotný výkop so zapojením troch kĺbov – členku, kolena a bedra. Cieľom bolo zapojiť väčší počet kĺbov a dosiahnuť tak sčítanie momentov
- posledná fáza je stabilizačná, slúži na návrat agenta do vzpriamenej polohy.

- ***Kop do lopty hranou chodidla***

V tomto prípade ide o kop do lopty hranou chodidla agenta. Nejde však o kop dopredu, ale do boku. Tento kop umožňuje agentovi nahrávku do strany bez nutnosti zdĺhavo sa nastavovať k lopte. Vzhľadom na možnosti a anatómiu agenta nejde o silný kop – na kop možno využiť iba jediný kĺb.

Tento pohyb má fázy:

- prvá fáza je založená na prvej fáze priameho kopu – agent sa prikrčí do polosedu, aby tak získal lepšiu stabilitu
- v druhej fáze sa agent naváži na jednu stranu, aby tak udržal rovnováhu po zodvihnutí opačnej nohy, ktoré nasleduje v ďalšej fáze. Zapája kĺby členkov, bedier, torzo sa mierne nakláňa dopredu kvôli stabilite a rovnováha sa vyvažuje aj rukou
- agent zapojením kĺbov bedra, kolena a členku vysunie nohu pred telo a zároveň mierne do boku, aby si tak prichystal vhodnú pozíciu na kop
- štvrtá fáza je samotný kop, ktorý realizujú bedrové kĺby a zároveň ho vyvažujú ruky (kĺby, ramena a lakt'a)
- posledná fáza je stabilizačná, slúži na návrat agenta do vzpriamenej polohy

1.1.2. Základné pohyby implementované vo finálnom produkte

➤ **Vstávanie**

• *Vstávanie zo sedu rozkročmo*

Tento pohyb vznikol kvôli potrebe bránenia v sede rozkročmo. Úpravou samotného sadnutia rozkročmo a jeho hlavnou myšlienkou bolo dostať obe nohy do základnej polohy bez použitia rúk len jednoduchým prinožením, pričom najväčším problémom bolo udržanie rovnováhy.

Fázy pohybu:

- prinožovanie do polostoja
- roznoženie, aby pokleslo ťažisko a zachovala sa rovnováha
- pokračovanie prinožovania až do úplného postavenia sa

➤ **Chôdza**

• *Chôdza do strany*

Táto chôdza bola implementovaná hlavne kvôli potrebe pohybu brankára. Implementovaná pre obe strany, pre každú dvomi spôsobmi, ktoré sa od seba odlišujú len tým, že jeden sa stáča mierne dozadu a druhý dopredu, teda robot v oboch prípadoch ide nie priamo doboku, ale v oblúku.

Využitelná je hlavne verzia so stáčaním dozadu, ktorá sa približuje reálnemu pohybu brankára

Fázy pohybu:

- pokrčenie nôh a prenesenie váhy na pravú nohu
- zdvihnutie ľavej nohy a jej natiahnutie čo najviac do strany
- dostúpenie na ľavú nohu a prenos váhy na ňu
- prisunutie pravej nohy

Tento pohyb je extrémne pomalý, ale pri pokusoch o zrýchlenie strácal stabilitu. Najväčší problém robí neúplné prenesenie váhy na nohu, ktorou sa robí úkrok. Napriek tomu je tento pohyb stabilný a v kontexte výkonu ostatných pohybov stále použiteľný.

• *Chôdza dozadu 2*

Vznikla z úkrokov a je pomalšia (drobnejšia) ako chôdza dozadu v prototypu. Výhodou tejto verzie chôdze dozadu je, že robot pri nej nepredkopáva, vďaka čomu sa hodí na presnejší pohyb v okolí lopty. Robot sa pri tejto chôdzi odráža zo špičiek.

Fázy pohybu:

- robot pokrčí nohy
- odrazí sa zo špičky, a tým sa posunie mierne šikmo dozadu
- dostúpi a vystrie nohy

Približné posunutie dozadu je o štvrt' chodidla. Kvôli týmto vlastnostiam nie je uvedený pohyb vôbec vhodný na prekonávanie väčších vzdialeností. Je tu veľký priestor pre vylepšenia. Výhodou tohto pohybu je pomerne veľká presnosť, keďže robot zachováva veľmi spoľahlivo priamy smer, kým nespadne.

➤ Otáčanie

Nanovo vytvorili otočenie o 90° do oboch strán, keďže verzia v prototypy nebola použiteľná. Vytvorili aj otočenie o 135° , 180° .

• *Otočenie o 90°*

Keďže pôvodný pokus vytvoriť tento pohyb na základe pozorovania jeho ľudskej verzie zlyhal, rozhodli sa vytvoriť jeho výrazne robotickú verziu, ktorá je prispôbená vlastnostiam a možnostiam fyziky servera. Pohyb je zimplementovaný do oboch smerov.

Fázy pohybu „pravej“ verzie:

- naváženie sa ľavú nohu tak, že hráč stojí len na nej
- zdvihnutie a pokrčenie pravej nohy
- pohyb v bedrovom kĺbe pravej aj ľavej nohy tak, že zvierajú pravý uhol
- polozenie pravej nohy a preváženie na ňu
- zdvihnutie, pokrčenie, prinoženie a vystretie ľavej nohy
- dokončenie pohybu celkovým narovnaním robota

Výsledný uhol otočenia je mierne väčší ako želaných 90° , avšak táto odchýlka je len niekoľkostupňová. Pohyb je oproti prototypu deterministický.

• *Plynulé drobné otáčanie - 5°*

Plynulé drobné otáčanie o 5° je cyklický pohyb, ktorý by mal hráč využívať v situáciách, keď sa potrebuje pomerne presne napríklad k lopte.

Fázy pohybu:

- agent sa mierne prikrčí a zloží ruky s cieľom zvýšenia stability
- naváži sa pootočením bedrových a členkových kĺbov na jednu zo strán
- mierne sa nadvihne a predsunie jednu z nôh
- polozenie nohy s miernym vychýlením, výsledkom je natočenie agenta o malý úsek
- finalizácia fázy slúži na postavenie agenta do východzej polohy

Agent sa počas jednej minúty dokázal otočiť približne o 130° , pričom sa ale neotáčal doslova, ale tvoril špirálu, čo ale nie je až také dôležité vzhľadom na podstatu, ktorou je otáčanie práve drobné otáčanie. Testovanie odhalilo stabilitu agenta. Trvanie pohybu je 2000 ms.

• *Plynulé drobné otáčanie - 20°*

Plynulé drobné otáčanie o 20° je cyklický pohyb, ktorý by mal hráč využívať v situáciách, keď sa potrebuje pomerne presne, po väčších úsekoch, natočiť, napríklad k lopte.

Je vďaka rozsiahlejším pohybom kĺbov menej stabilné ako plynulé otáčanie o 5° .

Fázy pohybu:

- agent sa mierne prikrčí a zloží ruky s cieľom zvýšenia stability
- podobne ako pri jemnejšom otáčaní sa naváži agent pootočením bedrových a členkových kĺbov na jednu zo strán

- výraznejšie nadvihnutie a predsunutie jednej z nôh
- polozenie nohy s vychýlením
- finalizácia fáza znovu slúži na postavenie agenta do východzej polohy

Výsledkom je natočenie agenta o úsek zodpovedajúci približne 20°. Agent sa pri testovaní otočil počas jednej minúty približne o 400°, tiež do špirály. V tomto prípade bol špirálový efekt výraznejší, nakoľko pri kročení môže aj vinou vnášania rôznych chýb zo strany servera spôsobiť menší či väčší posun.

➤ **Kop do lopty**

Implementovali ďalší kop a to priamy kop do lopty hranou chodidla.

Kop do lopty priamy hranou chodidla

Ide o kombináciu dvoch pohybov – priameho kopu a kopu do boku. Je to kop hranou chodidla, teda styčnou plochou hráča s loptou je hrana chodidla a lopta smeruje priamo.

Fázy pohybu:

- agent sa dostane do stabilizovanej polohy, ktorá mu umožňuje v nasledovnej fáze stáť na jednej nohe. Táto poloha spočíva v pokrčení kolien, nastavení torza a miernom upažení rúk
- naváže sa na jednu stranu (na ľavú alebo na pravú podľa toho, ktorou nohou agent kope)
- zdvihne a vytočí nohu tak, aby bolo možné kopnúť hranou chodidla
- samotný kop
- agent sa dostane opäť do stabilizovanej polohy.

Tento kop je určite slabší oproti ďalším dvom implementovaným kopom. Spôsobené je to najmä tým, že uhol vytočenia neumožní dostatočný náklon, a preto je tento pohyb vhodný skôr ako prihrávku, ako výkop na bránu. Zároveň však vďaka kopacej ploche má tento kop potenciál byť presnejším. Vďaka veľkej sile kopu je tento zároveň pomerne stabilný pohyb.

1.2. Testovanie pohybov tímu Androids

1.2.1. Bránenie

Bránenie sadnutím rozkročmo

Tento pohyb má slúžiť pre brankára. Jeho vykonanie má nastať v prípade, že strela smeruje do bránky v blízkosti brankára. V dokumentácii tímu Androids sa uvádza, že pri jeho testovaní nezaznamenali žiadnu chybu. Avšak pri našom testovaní pri vykonaní tohto pohybu

brankár väčšinou skončil na chrbte a v niektorých prípadoch aj v polohe tvárou k zemi. Ak agent skončil na bruchu nevedel sa navyše postaviť.

Pád na bok – fall_right

Ide o ďalší pohyb využiteľný brankárom. V prípade, že lopta smeruje do bránky a zákrok sadnutím rozkročmo by nestačil na zachytenie lopty, mal by sa použiť pohyb pádu na stranu. Pri testovaní tohto pohybu sa nestalo ani raz, že by sa agent prevrátil na brucho alebo chrbát. Avšak bolo by potrebné tento pohyb vykonať aj priamo v akcii, teda spolu s útočníkom, ktorý na brankára vystrelí, keďže tento test nebol vykonaný ani tímom Androids.

Otáčania

Otáčanie o 90° - turnleft90

Ide o otočenie agenta o uhol 90 stupňov (v skutočnosti je natočenie o niekoľko stupňov väčšie). Základom tohto pohybu je nadvihnutie jednej nohy, jej otočenie o 90°, polozenie na zem a následne priloženie druhej nohy. Samotný pohyb je stabilný avšak jeho vykonanie trvá takmer 5 sekúnd čo je pomerne veľa, takže by bolo potrebné tento pohyb zrýchliť bez toho aby sa narušila stabilita agenta.

Plynulé drobné otáčanie o 5° - turn_left_cont_5

Ide o krátke pohyby, ktoré natáčajú agenta o 5 stupňov. Agent nedokázal vykonať otáčanie na mieste, ale keďže tento pohyb sa bude používať len pre malé natočenia je tento pohyb postačujúci. Pri jeho testovaní nestratil agent stabilitu ani v jednom prípade.

Plynulé drobné otáčanie o 20° - turn_left_cont_20

Tento pohyb je podobný pohybu otáčaniu o 5° avšak uhol otáčania je v tomto prípade 20°. Otáčanie je trochu menej stabilné ale agent pri testovaní vykonal pohyb bez pádov. Aj pri tomto pohybe sa agent neotáča na mieste ale akoby do špirály.

Kopy

Kop do lopty priamy hranou chodidla – kick_straight_edge_r

Ide o kop vnútornou stranou chodidla. Pohyb vychádza z postavenia, v ktorom má agent stabilitu pri stoji na jednej nohe. Sila kopu je pomerne slabá, takže jeho využitie je len pre nahrávky. Táto nedostatočná razancia vyplýva hlavne zo slabej stability pri vykonávaní tohto pohybu. Pri jeho testovaní agent nepadol ani raz.

Priamy kop do lopty

Tento pohyb je kopnutím do lopty špicou chodidla. Jeho základom je prikrčenie sa, kedy sa agent dostane do stabilnejšej polohy pre vykonanie kopu. Pohyb pri testovaní pracoval správne a pád alebo zachytenie o podložku nenastal ani v jednom prípade. Agent sa pri vykonaní tohto pohybu vychýli o pár stupňov.

Vstávanie

Vstávanie z brucha

Test tohto pohybu bol spojený spolu s testom pomocného pohybu *pádvpred*. Pri ich realizácií agent padne tvárou k zemi a snaží sa následne postaviť. Všetky pokusy o vstanie pri testovaní boli úspešné.

Vstávanie z chrbta

Podobne ako vstávanie z brucha aj tento pohyb bol spojený s druhým pohybom, tentoraz *pádvzad*. Agent pri jeho vykonávaní spadne na chrbát a následne sa snaží postaviť. Aj v tomto prípade testy prebehli bez problémov.

Chôdza

Drobná pomalá chôdza dopredu

Táto chôdza neslúži na výrazné presúvanie ale len na presnejšie priblíženie sa k lopte. Dĺžka kroku je približne 1/5 chodidla a ide o chôdzu veľmi pomalú. Pri testovaní nedošlo k

problémom so stabilitou, avšak pri vykonávaní pohybu sa agent počas 3 minút vychýlil približne o 90°.

Drobná pomalá chôdza dozadu - walk_fine_back

Tím Androids uvádza v dokumentácii, že pohyb vznikol náhodne. Preto aj vizuálna stránka pohybu vyzerá zvláštne, keď agent pri jeho realizácii poskakuje a výrazne predkopáva. Pri testovaní bol pohyb stabilný a ani v jednom z prípadov agent nespadol.

Chôdza dozadu 2 - walkback2

Ide o veľmi malé úkroky smerom dozadu. Pri jej realizácii sa robot odráža od špičiek. Pohyb nie je moc stabilný, aj keď k pádom nedochádza. Avšak je pomerne pomalý a preto nie je vhodný na prekonávanie väčších vzdialeností. Tento pohyb by bolo vhodné pri ďalšej práci vylepšiť.

Drobčivá rýchla chôdza dopredu – walk_fine_fast2

Ide o pomerne rýchlu chôdzu smerom dopredu. Agent pri jej realizácii mierne naskakuje. Pohyb je stabilný, pri testoch nedošlo ani k jednému pádu.

Rýchla chôdza dopredu – walk_fine_fast1

Rovnako ako v predchádzajúcom prípade ide o pohyb smerom dopredu. Avšak v tomto prípade sa agent po niekoľkých krokoch rozkýve a následne spadne.

1.3.Reprezentácia pohybov – framework na tvorbu anotácie

Aby sa vedel hráč rozhodovať, ktorý pohyb naplánovať v danej hernej situácii, bezpodmienečne musí byť informovaný o tom, čo robí ktorý pohyb (aký bude mať vplyv jeho naplánovanie na budúcnosť, ktorý z pohybov je vhodné vybrať v danej situácii). Preto je potrebné vyrobiť anotáciu pre pohyby, vďaka ktorej bude možné automatizovať ich plánovanie.

Požiadavky na anotácie:

1. strojová spracovateľnosť (počítač musí vedieť robiť rozhodnutia na základe anotácie)
2. obšírnosť (čím viac informácií hráč má, tým lepšie sa vie rozhodovať)
3. automatizovateľnosť (vlastnosti pohybov by malo byť možné vyplňať automaticky, analýzou ich reálneho behu, s minimálnym príspevom človeka)
4. zrozumiteľnosť (je možné, aby im porozumel aj človek – mohol ich vytvárať, čítať a editovať)
5. všeobecnosť (množina atribútov reprezentujúcich pohyby by mala byť pre všetky pohyby rovnaká – táto vlastnosť zaručí jednoduchosť práce s anotáciami)

Počas analýzy sme identifikovali niektoré základné atribúty, ktoré musí anotácia obsahovať:

- *typ pohybu* – hráč si na základe neho môže pre danú situáciu vyselektovať množinu teoreticky použiteľných pohybov (napr. ak sa potrebuje presunúť z jedného miesta na druhé, bude vyberať spomedzi pohybov typu „chôdza“, v inej situácii typ „kop“).
- *zmena pozície a natočenia hráča* počas trvania pohybu – táto znalosť je potrebná pri výbere pohybu, ak sa hráč potrebuje presunúť alebo otočiť. Môže byť rozhodujúca aj za okolností, že posunutie či otočenie hráča v danej situácii nie je vítaná vlastnosť pohybu.
- *dĺžka trvania pohybu* – na jej základe si môže vybrať napr. najrýchlejší dostupný pohyb s potrebnými vlastnosťami.
- *pravdepodobnosť pádu* počas vykonávania pohybu - na jej základe si môže vybrať napr. najspoľahlivejší dostupný pohyb s potrebnými vlastnosťami.
- *predpoklady* – hráč si vyberie taký pohyb, ktorý nadväzuje na jeho aktuálny stav.
- *stav po vykonaní pohybu* – umožňuje hráčovi plánovanie ďalších pohybov do budúcnosti, skladanie pohybov.
- *sledovanie lopty* - na základe vzdialenosti od lopty sa hráč rozhodne, či do nej môže kopnúť a ak áno, tak ako ďaleko a v ktorom smere ju dokáže kopnúť.

Je dôležité si uvedomiť, že rovnaký pohyb je vykonaný vždy trochu inak – preto je potrebné pre popísanie pohybu používať priemerné hodnoty z viacerých behov daného pohybu.

Pri počítaní zmeny pozície hráča v dôsledku vykonania pohybu je dôležité si uvedomiť, že hráč vie počítať zmenu svojej pozície iba z pohľadu ihriska, nás ale zaujíma zmena pozície z

pohľadu hráča. Preto je potrebné túto hodnotu počítať na základe kombinácie informácií o zmene pozície na ihrisku a otočenia hráča pred vykonaním pohybu.

Na druhú stranu potom hráč bude musieť vedieť pri plánovaní pohybu prepočítať na základe anotácie a svojho otočenia, aký bude mať vplyv vykonanie daného pohybu na jeho pozíciu v rámci ihriska.

1.4. Analýza bakalárskej práce Marcela Kantu

1.4.1. Zameranie bakalárskej práce

Bakalárska práca je zameraná na stabilizáciu hráča, skúmanie nižších schopností existujúceho hráča a návrh vylepšení nevyhnutných k stabilizácii. V práci je navrhnuté úmyselné umiestňovanie ťažiska tak, aby hráč získal lepšiu rovnováhu, a tak bola znížená pravdepodobnosť pádu. Riešenie spočíva v pohybe, ktorým dokáže hráč predchádzať pádu.

1.4.2. Návrh riešenia

Detekcia pádu – princíp je využitie ťažiska, alebo akcelerometra vzhľadom k pozícii nôh vo vodorovnej rovine. Akcelerometer meria zrýchlenie vzhľadom na gravitáciu, približne $(0,0,g)$ – nie je to presné, pretože tam vznikajú odchýlky vnášané serverom. Túto hodnotu v absolútnych súradniciach pripočítal k ťažisku. Pri výpočte posunu vychádzal zo vzorca $s = vt + at^2$, hráč sa posúva za určitú časovú vzorku. A teda vypočítal predpokladanú pozíciu ťažiska za určitý čas.

Naivná detekcia stability – v priebehu testovania zistil, že táto metóda nie je vhodná, počítal tu s priemerom súradníc stredu chodidiel vo vodorovnej rovine a vektorom posunutia od ťažiska k tomuto miestu.

Metóda stabilnej oblasti – spolieha sa na predpovedanie pozície ťažiska s ohľadom na aktuálny stav stabilnej oblasti – priestore pod nohami vo vodorovnej rovine v absolútnych súradniciach.

1.5. Analýza modelu sveta

Model sveta je obsiahnutý v komponente Models. Kde sa nachádzajú 3 časti, a to AgentModel, EnvironmentModel a WorldModel. V triede AgentModel sa počítajú pozície a rotácie kĺbov, v triede WorldModel sa počítajú pozície ostatných hráčov a lopty a v triede

EnvironmentModel sa ukladá mód hry, čas a verzia servera. Na analýzu bola nutná prehliadka zdrojového kódu, ktorý nebol vhodne okomentovaný.

1.5.1. Triedy modelu sveta

➤ AgentModel.java

obsahuje určovanie pozície hráča a stranu hráča (člen tímu, súper)

metódu na výpočet pohybu vektora okolo vektorov a osí, výpočty sú vykonávané podľa teórie vektorových pohybov nájdené na internete

obsahuje GET na získanie informácií či stojí, je na zemi, leží na chrbáte, leží na bruchu, o natočení a pozícii a SETY na nastavenie pozície a rotácie

➤ AgentPositionCalculator.java

vypočíta aproximáciu aktuálnej pozície agenta na základe zástavy, ktorú vidí. Trieda predpokladá, že rotácie agenta sú správne vypočítané. Výpočet sa vykonáva normalizáciou a odpočtom od pozície zástavy. Keď vnímaná relatívna pozícia je pridaná do známej pozície vlajky, je možné vypočítať správnu aproximáciu našej pozície.

➤ AgentRotationCalculator.java

počíta sa natočenie agenta podľa fixných bodov – dokáže odvodiť natočenie agenta z 3 videných zástav, z 3 zástav si určí, ktoré dve z nich použiť na vytvorenie osi Y, a ktorú použiť ako doplnkový bod na vytvorenie osi Z, kolmej na os Y.

➤ DynamicObject.java

Trieda určuje polohu dynamických objektov, zvyčajne lopty, chýba tu vypočítanie pozície súperov

➤ EnvironmentData.java

ukladá statické informácie o stave súčasného sveta okolo agenta – herný čas, simulačné časy, režim prehrávania a verziu servera. Verzia servera je stanovená v ./scripts/config/settings.rb – momentálne 0.6.4

➤ FixedObjects.java

tu každá vlajka uchováva informáciu o svojej pozícii – určuje sa tak veľkosť ihriska

➤ KalmanAdjuster.java

nastavuje vnímané súradnice lopty a zástavy na ihrisku použitím kalmanvého filtra, čím sa znižuje chybovosť, vnášaná serverom, pre ďalšie výpočty

➤ Player.java

trieda ktorá definuje agenta ako hráča v hre – nič nie je spravené

➤ WorldModel.java

ukladanie objektov na ihrisku, ukladá ich pozíciu, rýchlosť a priemernú pozorovanú rýchlosť sem treba dorobiť predvídanie tímu a oponentov a učenie kto je z ktorého tímu

➤ Prophecy.java

uchováva pravdepodobný stav v danom čase v budúcnosti

➤ Prophet.java

výpočet najpravdepodobnejšej udalosti v danom čase, treba tu dorobiť predpovedací modul, momentálne je to iba v stave experimentu a na konci projektu si neboli istý, či to funguje správne.

Vykonanie akcie stabilizácie – princíp vychádza z identifikovania vektora stabilizácie a vykonania pohybu, ktorým sa ťažisko pohne o vektor stabilizácie do stabilnej oblasti. Chcel použiť na ovplyvnenie ťažiska ruky, no mali malú hmotnosť a tak nízky vplyv. Zistil, že trup silno ovplyvňuje pozíciu ťažiska (trup nemá kĺby ale je ovplyvňovaný stehnami a bedrovými kĺbmi). Zistil, že pre stabilizáciu platí, čím ďalej agent padá, tým väčší protipohyb je potrebný. Ďalej, že stabilizácia by mala byť transparentná voči vyššej logike, a teda vyššia logika by mala prijímať vstupy ako keby bez stabilizácie, navrhol riešenie vytvorenia medzivrstvy medzi reálnym natočením a natočením. Reálne to vyriešil tak, že v každom kĺbe ukladal aj stabilizačnú odchýlku.

Postup stabilizácie – vyššia logika navrhne natočenie kĺbov a následne stabilizačný modul analyzuje či je nutné stabilizovať, alebo nie. Ak modul zistí, že má dôjsť k pádu, aktualizuje stabilizačnú odchýlku. Pri testovaní odhalil nedostatok, a to že mu vznikala príliš veľká odchýlka, preto vytvoril 3 stavy hráča: nestabilný (zvyšuje sa odchýlka), medzistav (agent nepadá ale odchýlku nie je možné vynulovať), stabilný (vynulovanie odchýlky).

Vizualizácia – zaoberal sa aj vizualizáciou v matlabe, hlavne zobrazovaním počítaných bodov a vektorov, napríklad súradnice agenta, alebo pomocné vektory (absolútne aj relatívne).

Matematická knižnica – vytvoril vlastnú matematickú knižnicu, pretože riešenie s ktorým pracoval (RoboKopy) mali vytvorenú knižnicu na mieru, slabo okomentovanú a bez dokumentácie.

1.5.2. Zhodnotenie

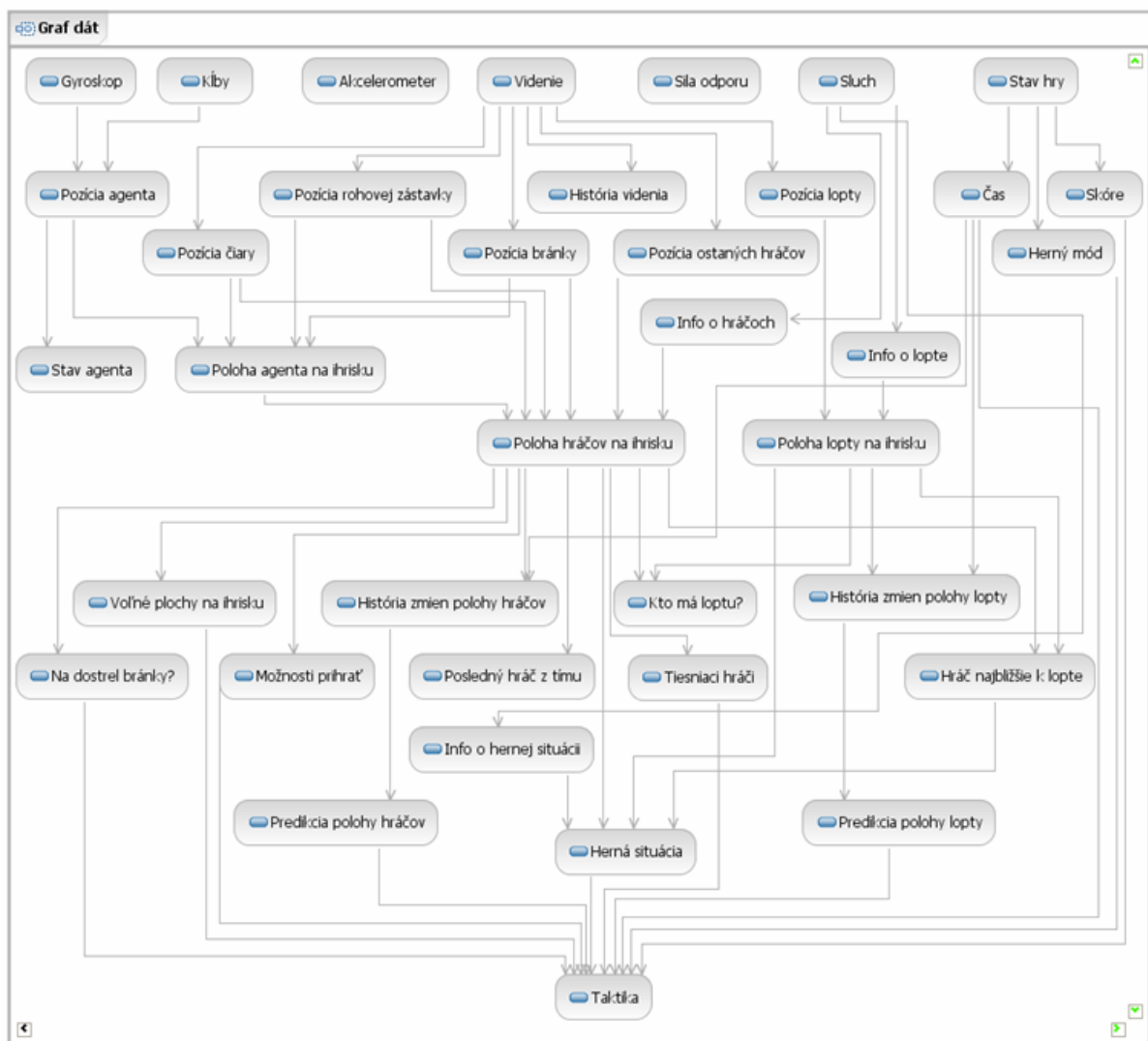
Pri práci musel čeliť viacerým problémom, ako zmena servera počas semestra, alebo chyby v agentoch. Implementoval modul stabilizácie s detekciou nestability, avšak pri konečnom testovaní zistil, že algoritmus je prakticky nepoužiteľný, agent často padal.

1.6. Model sveta

Táto podkapitola pojednáva o Modely sveta, o dátach, ktoré by sa pomocou výpočtu o svete agent mohol dozvedieť, aké dáta aktuálne dokáže zistiť a aké ďalšie by bolo vhodné doplniť.

1.6.1. Graf dát

Graf dát predstavuje informácie, ktoré dokáže agent o svete získať. Informácie sú na rôznych stupňoch abstrakcie. Najnižšiu vrstvu predstavujú tie, ktoré prichádzajú zo servera a predstavujú dáta, ktoré agent dostáva zo snímačov. V grafe sú to uzly, do ktorých nevchádza žiadna hrana.



Obrázok č.1: Graf dát

1.6.2. Uzly v grafe

Gyroskop

Údaje z GyroRate perceptoru v tvare:

(GYR (n <name>) (rt <x> <y> <z>))

Kĺby

Údaje z perceptorov typu HingeJoint a UniversalJoint v tvare:

(HJ (n <name>) (ax <ax>))

(UJ (n <name>) (ax1 <ax1>) (ax2 <ax2>))

Sila odporu

Údaje z perceptorov ForceResistance v tvare:

(FRP (n <name>) (c <px> <py> <pz>) (f <fx> <fy> <fz>))

Sluch

Údaje z perceptorov Hear v tvare:

(hear <time> self/<direction> <message>)

Akcelerometer

Údaje z perceptorov Accelerometer v tvare:

(ACC (n <name>) (a <x> <y> <z>))

Videnie

Údaje z perceptoru Vision v tvare:

(See +(<name> (pol <distance> <angle1> <angle2>))

+ (P (team <teamname>) (id <playerID>) +(<bodypart> (pol <distance>
<angle1><angle2>))))

+ (L (pol <distance> <angle1> <angle2>) (pol <distance> <angle1> <angle2>))))

Pozícia bránky

Relatívna pozícia bránky vzhľadom na agenta

Pozícia rohovej zástavky

Relatívna pozícia rohovej yzástavky vzhľadom na agenta

Pozícia lopty

Relatívna pozícia lopty vzhľadom na agenta

Pozícia ostatných hráčov

Relatívna pozícia hráčov vzhľadom na agenta

Pozícia čiar

Relatívna pozícia čiar vzhľadom na agenta

Stav hry

Údaje z perceptoru GameState v tvare:

(GS (t <time>) (pm <playmode>))

Herný mód

Aktuálny herný mód (priamy kop, penalta, rohový kop...).

Poloha agenta na ihrisku

Poloha agenta na na ihrisku. Absolútna poloha.

Pozícia agenta

Pozícia agenta v akej sa nachádza. To je poloha ťaziska ajednotlivých končatín.

Stav agenta

To v akej polohe je agent a čo sa s ním deje. Aký pohybvykonáva. Poloha voči lopte, bránke.

Či je tiesnený súpeom. Či sa nachádza v blízkosti lopty, alebo na dostrel braky.

Poloha lopty na ihrisku

Poloha lopty a jej súradnice v stanovenej súradnicovejsústave. Absolútna poloha na ihrisku

Poloha ostatných hráčov na ihrisku

Poloha ostatných hráčov na hracej ploche v stanovenejsúradnicovej sústave. Absolútna poloha na ihrisku.

Herná situácia

Aktuálna herná situácia odohrávajúca sa na ihrisku. Útočenie, obrana...

Hráč najbližšie k lopte

Hráč, ktorý je aktuálne najbližšie k lopte.

Voľné plochy na ihrisku

Plochy na ihrisku, kde sa nenachádzajú protihráči.

Posledný hráč z tímu

Hráč, ktorý predstavuje brankára.

Čas

Aktuálny čas.

Skóre

Aktuálny stav zápasu.

História zmien polohy lopty

Polohy lopty za posledných x sledovaných momentov.

Predikcia polohy lopty

Predpokladaná poloha lopty v nasledujúcom momente.

História zmien polohy hráčov

Polohy hráčov za posledných x sledovaných momentov.

Predikcia polohy hráčov

Predpokladaná poloha hráčov v nasledujúcom momente.

Taktika

Taktika, akú budú hráči používať na základe situácie a stavu hry.

Kto má loptu?

Ktorý hráč má loptu.

Môžnosti prihrávky

Všetky aktuálne možnosti na prihrávku spoluhráčovi.

Tiesniaci hráči

Všetci hráči v nebezpečnej vzdialenosti od agenta.

Na dostrel branky?

Overenie či je konkrétny agent vo vzdialenosti odkiaľ dokáže dostreliť na bránu.

Info o hráčoch

Informácie prijaté sluchovým perceptorom týkajúce sa polohy hráčov.

Info o lopte

Informácie prijaté sluchovým perceptorom týkajúce sa polohy hráčov.

Info o hernej situácii Informácie prijaté sluchovým perceptorom týkajúce sa hernej situácie..

Údaje o type a formáte prichádzajúcich dát zo servera sú zo stránky [\[2\]](#), tiež je tam možné nájsť detailnejšie informácie k jednotlivým efektorom a popis k formátu dát.

1.7. Analýza diplomovej práce: Bc. Ján Kolesár – Simulovaný robotický futbal – automatizované učenie

Študent v tejto práci skúmal možnosti využitia evolučných algoritmov pre naučenie hráča základným nízkourovňovým schopnostiam. Upravoval existujúci evolučný algoritmus hráča tímu Neurotics. Výsledkom práce bolo zistenie, že tento princíp neprináša dobré výsledky – hlavným problémom bolo vytvorenie vhodne fitness funkcie. Agent využívajúci vzniknuté dáta používal verziu serveru 0.6.0.

Fitness funkcia ktorú študent navrhol vychádzala z týchto vlastností: [\[3\]](#)

1. Rozdiel maximálnej dosiahnutej výšky kamery a výšky kamery vo vzpriamenej polohe.

2. Vzdialenosť v rovine XY, o ktorú sa posunul od pôvodnej polohy.
3. Uhol, o ktorý sa líšil od zvislej polohy.
4. Čas za ktorý sa do tejto výšky dostal ako dlho v nej zotrval (či potom nespadol)

Samotná evolúcia prebiehala priamo v prostredí simulačného servera, čo spôsobovalo mnohé problémy ovplyvnené častým padaním servera a jeho pomalým výkonom. Kvôli času potrebnému na evolúciu sa študent zamerlal len na statické pohyby.

Ďalším z problémov ktoré sa vyskytly bolo uviaznutie hráča v lokálnych extrémoch, ako napríklad polohy sedu. Riešený bol najmä dynamickou zmenou pravdepodobnosti mutácie.

Významnými problémami prostredia, ktoré študent popísal boli: [\[3\]](#)

- Chyba v detekcii kolízií hráča s hracou plochou
- Rozpadnutie hráča
- Nestabilná dĺžka cyklu
- Šum
- Reálny čas simulácie

Niektoré z týchto problémov, najmä prvé dva, boli odstránené alebo zmenšené v súčasnej verzii servera (0.6.5). Iné však nie sú chybami, ale zámernými vlastnosťami simulačného prostredia.

Okrem možnosti vytvoriť hráča evolúciou od základu skúšal študent aj možnosť začatia evolúcie na existujúcom hráčovi, ktorého by sa tak mohlo podariť vylepšiť. Tu však výsledky neboli o nič lepšie.

1.8. Možnosti využitia paralelizácie RoboCup servera

1.8.1. Úvod

Pre dosiahnutie čo najlepších vlastností hráča bude potrebné spustiť veľký počet simulácií, čo naráža na výkonnostné obmedzenia samotných agentov, ale aj servera využívaného v simulovanom RoboCupe.

Na fakulte je k dispozícii pomerne výkonný výpočtový cluster, momentálne však server ani agenti nie sú v stave, kedy by sa tam dali jednoducho nasadiť.

Prvým krokom na zistenie, na ktorú oblasť sa oplatí v tomto smere zamerať bolo zmeranie výkonu servera bežiaceho „naprázdno“ aj s rôznym počtom pripojených agentov. Meranie prebiehalo na pomerne slabom virtuálnom serveri pridelenom fakultou, s 3 virtuálnymi 2.2 GHz jadrami a 2 GB RAM.

Pre riadenie servera som si vytvoril jednoduché webové rozhranie umiestnené na <http://robo.holak.net/>

Umožňuje spúšťať/zastavovať simulačné prostredie, pozerat' si jeho výstup (pre kontrolu pripájania/odpájania agentov) a sledovať vyt'áženie systému ako výstup príkazu top.

Vyt'áženie procesora pri pripojení viacerých agentov na server stúpa lineárne, v prostredí kde medzi sebou hráči príliš neinteragujú (tj. server zrejme nestrávi toľko času detekciou kolízií).

SimSpark v súčasnosti obsahuje experimentálnu podporu behu simulačných cyklov vo viacerých vláknach, no tie sú používané len na umožnenie súčasného behu fyzikálnych výpočtov a riadiacich cyklov. [6]

1.8.2. Možnosti

Ako je uvedené v [5], jednu z výkonnostne najnáročnejších častí simulačného prostredia tvorí simulácia fyzikálnych vlastností sveta. Tá je vykonávaná pomocou knižnice Open Dynamics Engine (ODE). Skupina autorov [4] sa zaoberala možnosťami paralelizácie niektorých vlastností tejto knižnice, konkrétne detekcie kolízií a problému lineárnej komplementarity. Najmä prvá z uvedených by mohla výrazne napomôcť zrýchleniu simulácií. V plánoch pre ďalší vývoj servera je explicitne spomenuté využitie viacvláknovej verzie ODE, s poznámkou že určitá časť podpory už je implementovaná – nie však v súčasne vydanéj stabilnej verzii. Sú však spomenuté aj plány pre integráciu iných fyzikálnych knižníc – Bullet a PhysX, takže či sa bude vývoj servera budúcnosti uberať týmto smerom nie je isté. Obe tieto knižnice obsahujú podporu pre paralelné výpočty pomocou CPU aj s využitím GPU (napr. Bullet dokáže využiť CUDA, OpenCL, DirectCompute).

1.8.3. Problémy

Problémom je aj nedostatok testerov pre tieto vlastnosti, nakoľko testovanie a ladenie viacvláknových programov je všeobecne výrazne náročnejšie než u sekvenčných programov. To dokazuje aj fakt, že len pomerne nedávno boli opravené race condition chyby vo viacvláknových častiach súčasného servera.

V situácií kedy nemáme dostatočné paralelné (tj. paralelizované výkonnostne najnáročnejšie časti) simulačné prostredie nemá príliš veľký význam zaoberať sa paralelizáciou samotných

agentov, nakoľko jednotliví agenti nie sú výkonnostne tak nároční, aby sa vynaložené úsilie oplatilo – najmä v situácií keď na jednom stroji beží len jeden agent.

1.8.4. Záver

Určitý potenciál pre výrazné zrýchlenie simulácií pomocou paralelizácie fyzikálnych výpočtov rozhodne existuje, a v súčasnej dobe na ňom autori servera pracujú – je jednou z priorít pre roky 2011 a 2012. Jeho reálne dokončenie a nasadenie sa však nedá v najbližšej dobe očakávať a preto nemá zmysel zaoberať sa ním z praktického hľadiska pri vývoji agenta.

1.9. Vytvorenie skriptu na offline sťahovanie údajov

Keďže na jednom zo stretnutí k tímovému projektu sme zistili problém s prístupom na server tímu zo školskej siete, bol vytvorený skript, ktorý automaticky stiahne súčasnú verziu obsahu wiki vrátane všetkých odkazovaných dokumentov pre možnosť offline prezerania.

Okrem wiki sťahuje aj vybrané časti zo systému Redmine – novinky, úlohy a backlogy oboch tímov. Využíva na to program httrack. Peter Holák bol poverený spustením skriptu a prinesením offline obsahu na všetky stretnutia k tímovému projektu.

2. Návrh

2.1. *Reprezentácia pohybov – framework na tvorbu anotácie*

2.1.1. *Reprezentácia údajov*

Na základe uvedených požiadaviek sme sa rozhodli ukladať anotácie k pohybom v samostatných súboroch typu XML. Takto dostupné anotácie je možné pomerne jednoducho spracovávať a meniť tak človekom ako i strojom. V prípade potreby je možné takúto reprezentáciu anotácie vložiť priamo do XML súboru pohybu. V tabuľke 1 sú popísané všetky podstatné atribúty anotácie.

2.1.2. *Proces automatizácie vytvárania anotácie*

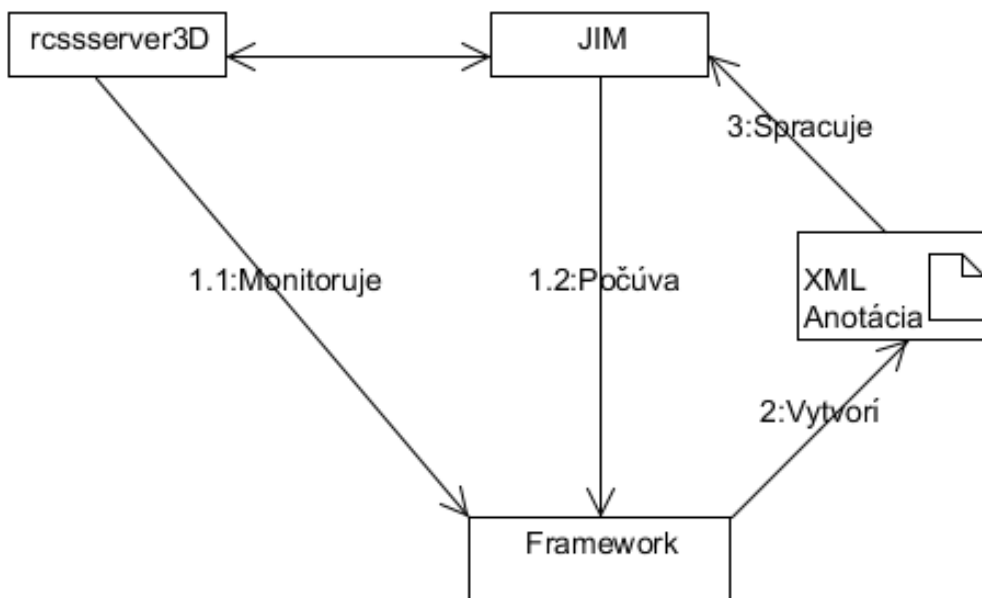
Napriek tomu, že človek musí vyplniť niektoré logické atribúty anotácie (typ pohybu, poznámka), väčšina je spracovateľná automaticky a naopak, pre človeka je náročné takéto hodnoty zisťovať (napr. priemerné otočenie hráča počas pohybu). Preto navrhujeme nasledovný spôsob automatizácie vytvárania anotácie (Obrázok 1) zložený z 3 krokov:

1. *Zbieranie údajov:*

1. Framework vie na základe monitorovania servera presné údaje o stave ihriska (kde je hráč, ako je otočený a pod).
2. Hráč plánuje svoje pohyby, preto vie presne povedať, kedy začal a kedy skončil vykonávanie akého pohybu.

2. *Vyhodnocovanie údajov:* Framework takéto informácie z oboch zdrojov dokáže spájať v zmysle zisťovania vlastností jednotlivých pohybov a exportovať na základe nich súbory s XML anotáciou pohybu (priemerné hodnoty pre každý použitý pohyb).

3. *Spracovanie anotácie:* Hráč dokáže takto vytvorenú XML anotáciu využiť pri plánovaní pohybov.



Obrázok č.1: Proces spracovania údajov

2.1.3. Vysvetlivky k XML reprezentácii anotácie

Tabuľka č.1: Vysvetlivky pre XML anotáciu pohybu

Značka	Význam
<anotation>	Koreňová značka
<name>	Meno pohybu, na ktorý sa anotácia vzťahuje
<checksum>	Kontrolná suma vzťahujúca sa na XML anotovaného pohybu, aby sa zabránilo nekonzistentnosti anotácie a pohybu v dôsledku jeho zmeny. Používaný algoritmus SHA-1.
<type>	Typ pohybu, kategória. Môže nadobúdať hodnoty: kick, walk, fall, look, stand_up, rotation, other. Môže mať aj viacero z týchto hodnôt (napr. ak sa jedná o zložitejší pohyb).
<description>	Predstavuje opis pohybu, obsahuje značky <duration>, <rotation>, <move>, <fall>, <ball_move>
<duration>	Priemerná dĺžka trvania pohybu v milisekundách.
<rotation>	Priemerná zmena natočenia hráča počas vykonania pohybu. Sú v ňom vnorené značky určujúce hodnoty otočenia po jednotlivých osiach v stupňoch. (Hodnoty otáčania okolo osi X a Y sú zatiaľ nepovinné, keďže ich využitie je momentálne nepravdepodobné)
<move>	Priemerná zmena polohy hráča počas vykonania pohybu. Sú v ňom vnorené značky určujúce hodnoty posunu po jednotlivých osiach z

	pohľadu hráča v metroch.(Hodnota posunu hráča po osi Z je zatiaľ nepovinná, keďže jej využitie je momentálne nepravdepodobné)
<fall>	Priemerný počet pádov hráča počas 100 vykonaní daného pohybu.
<ball_move>	Zmena polohy lopty po vykonaní pohybu. Sú v ňom vnorené značky určujúce hodnoty posunu po jednotlivých osiach z pohľadu hráča v metroch.
<note>	Poznámka autora stručne opisujúca daný pohyb v ľudskej reči.
<preconditions>	Obsahuje informácie o predpokladoch pre vykonanie daného pohybu (v akom stave sú kĺby a aká je poloha hráča na začiatku pohybu). Obsahuje značky <joints> a <lying>
<end_state>	Obsahuje informácie o stave hráča po vykonaní daného pohybu. Obsahuje značky <joints> a <lying>
<joints>	Obsahuje značky označujúce jednotlivé kĺby a hodnotu ich natočenia. Prednastavená hodnota pre každý kĺb je 0. Značky pre jednotlivé kĺby a ich ohraničenia sú kompatibilné so značkami použitými pre pohyb hráča (v prípade potreby dostupné v XSD súbore)
<lying>	Informácia o polohe hráča. Môže nadobúdať hodnoty: false, on_back, on_belly, on_side.
<ball_position>	Poloha lopty z pohľadu hráča vyjadrená vzdialenosťou na jednotlivých osiach v metroch.
<min><max><avg>	Značky obsahujú rozptyl a priemernú hodnotu niektorých atribútov pohybu.

2.2. Návrh na vylepšenie modelu sveta

Táto kapitola pojednáva o jednotlivých častiach modelu sveta, ktoré by bolo vhodné doplniť alebo vylepšiť v prvom rade.

2.2.1. Vylepšenie počítania polohy hráča

Ako vyplýva z predchádzajúcej kapitoly, momentálne nedokážeme určiť polohu sveta na 100% presne, preto by bolo vhodné výpočet tejto polohy overiť pomocou výpočtu polohy hráča voči väčšiemu počtu statických prvkov na ihrisku. Server posiela aj informáciu a čiary, ktorú agent vidí a to začiatkový bod a koncový bod, ale začiatkový a koncový bod nie kompletnej čiary, ale časti čiary, ktorú vidí vo svojom 120 stupňovom uhle. Z toho však vyplýva aj možný problém, že určenie, na ktorú čiaru sa pozerá, nemusí byť úplne triviálne, ale malo by sa dať určiť z natočenia hráča.

2.2.2. *Počítanie polohy ostatných hráčov*

Vzhľadom k tomu, že futbal je tímová hra, bolo by dobré, aby agent vedel aj o ostatných hráčoch na ihrisku. Túto informáciu mu najlepšie poskytne informácia o ich polohe na hracej ploche. Tieto informácie sa môže dozvedieť buď zo zrakového senzora, ale bolo by tiež zaujímavé ak by si hráči dokázali túto informáciu podať aj pomocou reči.

2.2.3. *Prerobenie zrakového perceptoru z hrude do hlavy*

Dôvod tejto zmeny je celkom prostý a to, že na novom serveri bol premiestnený zrakový perceptor z hrude do stredu hlavy, preto tomu treba prispôbiť aj agenta, aby nedochádzalo k neželaným odchýlkám už pri získavaní informácií.

2.2.4. *Odstránenie chýb vnášaných serverom*

Ako bolo spomínané vyššie, tým, že server vnáša do komunikácie šum, je vhodné použiť nejakú metódu, ktorá by dokázala minimalizovať prichádzajúce chybné údaje na minimum, či už to je pomocou kombinácie viacerých objektov pri výpočte polohy, ktoré na ihrisku vidí agent, ale tiež by bolo zaujímavé použiť iné metódy ako napríklad kalmanové filtre.

2.2.5. *Predikcia pohybu*

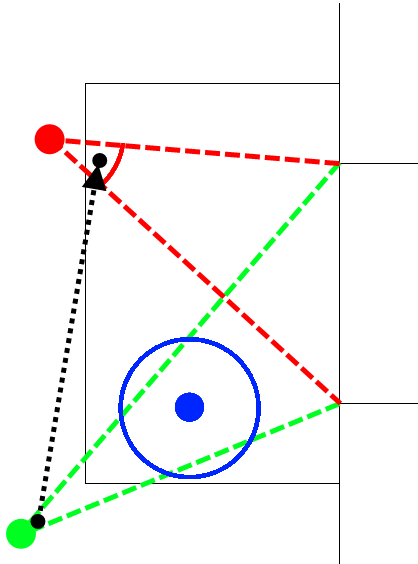
Pre lepšiu schopnosť agenta zistiť čo sa na ihrisku deje by bolo vhodné vylepšiť existujúci dynamický objekt, ktorý počíta rýchlosť len na základe súčasnej a predchádzajúcej polohy a to buď pridaním väčšieho počtu predchádzajúcich polôh, alebo skvalitnením výpočtu.

2.2.6. *Počítanie vzdialenosti lopty od hráča*

Táto informácia má možné využitie pri taktike, aby nedošlo k situácii, že sa všetci hráči rozbehnú za loptou, ale iba ten čo je najbližšie k nej. Táto informácia by mohla vypočítať z pozície agenta, pozície lopty a pozíí ostatných agentov.

2.3. Taktické pohyby

2.3.1. Prihrávka na gól



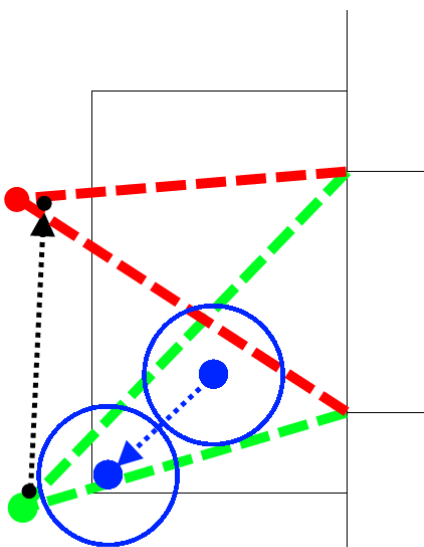
a loptu odkopne spoluhráčovi.

Najprv sa vypočíta, ktorý z hráčov útočiacich na bránu má „lepší výhľad“ na bránu, teda ktorý z nich má väčšiu možnosť skórovať gól. Do úvahy treba taktiež brať tzv. „pole pôsobnosti“ brankára, t.j. nejaký kruh okolo brankára, v ktorom môže zasiahnuť či už pádom, posadením sa alebo inak.

Ak má spoluhráč (hráč bez lopty) lepší výhľad, tak sa zistí jeho presná pozícia, vzájomná vzdialenosť medzi ním a hráčom s loptou a vypočíta sa rýchlosť a smer kopu do lopty.

Hráč s loptou sa k nej postaví v správnom smere

2.3.2. Vyčkávanie na prihrávku



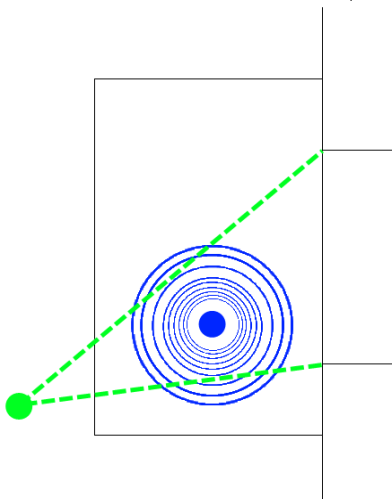
Postup je rovnaký ako v predchádzajúcom prípade (prihrávka na gól), no okrem toho sa vypočíta vzdialenosť medzi hráčom s loptou a brankárom.

Hráč chvíľu počká, znovu odmeria túto vzdialenosť a zistí, či sa brankár hýbe smerom k nemu.

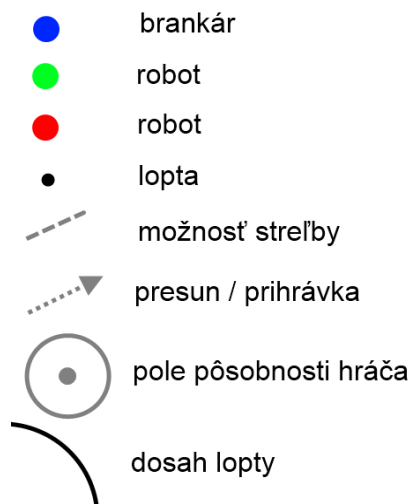
Ak áno, tak počká, kým sa táto vzdialenosť nezastaví alebo kým brankárovi „pole pôsobnosti“ nie je „dostatočne“ blízko k trajektórii možnej prihrávky, čiže pokiaľ nepretína túto trajektóriu.

Ak je to dostatočne blízko, tak prihrá loptu.

2.3.3. Strela na bránu



Legenda:



3. Použitá literatúra

[1] Dokumentácia tímu Android. Dostupné na internete:

http://labss2.fiit.stuba.sk/TeamProject/2010/team05is-si/files/dok_produk_t_verfinzs_20101214.pdf

[2] Detailné informácie k efektorom. Dostupné na internete:

<http://simspark.sourceforge.net/wiki/index.php/Effectors>

[3] Kolesár Ján, Bc., Simulovaný robotický futbal – automatizované učenie. Slovenská technická univerzita v Bratislave, 2009

[4] Michelle Goodstein, Michael Ashley-Rollman, Paul Zagieboylo , Parallelizing the Open Dynamics Engine. Dostupné na internete:

http://www.cs.cmu.edu/~mpa/ode/final_report.html

[5] Maintenance Committee 2011. Dostupné na internete:

http://simspark.sourceforge.net/wiki/index.php/Maintenance_Committee_2011

[6] Simulation Update Loop. Dstupné na internete:

http://simspark.sourceforge.net/wiki/index.php/Simulation_Update_Loop