



RoboCup S - Nové stratégie

Projektová dokumentácia

verzia 0.3



História zmien

V tab. 1 je história zmien, ktoré sa udiali v dokumente.

Tab. 1: História zmien dokumentu

Dátum zmeny	Verzia dokumentu	Popis zmeny	Autor
7.11.2007	0.1	Zosumarizovanie všetkých dokumentov do jedného dokumentu	Všetci
12.11.2007	0.2	Formátovanie dokumentu, štandardizácia	Ján Kohut, Ladislav Borženský, Marek Koperdák
14.11.2007	0.3	Kompletizácia dokumentu, konečná dokumentácia	Ján Kohut, Ladislav Borženský, Marek Koperdák



Tím

Bc. Ladislav Borženský	lborzensky zavináč gmail bodka com
Bc. Peter Brtáň	pbrtan zavináč gmail bodka com
Bc. Ján Kohút	jan.kohut84 zavináč gmail bodka com
Bc. Marek Koperdák	koperdak zavináč gmail bodka com
Bc. Vladimír Janov	vjanov zavináč gmail bodka com
Bc. Martin Petráš	petras.martin zavináč gmail bodka com

Webová stránka tímu: <http://www2.dcs.elf.stuba.sk/TeamProject/2007/team16is-si/>

Tímový mailinglist: jahodovi-princovia zavináč googlegroups bodka com



Anotácia

Predkladaný dokument obsahuje analýzu problému turnaja v robotickom futbale, špecifikáciu požiadaviek a hrubý návrh tímu Jahodoví princovia na vylepšenie hráča tímu Gang Of Six. Jednotlivé časti dokumentu boli vytvorené tak, aby spĺňali požiadavky kladené na výsledný produkt prvej fázy odovzdávania v rámci predmetu. Tvorba softvérového systému v tíme. V časti analýza problému, tím Jahodoví princovia zanalyzoval server simulujúci zápasy a tímy, ktoré sa venovali tejto problematike, či už na pôde Slovenskej technickej univerzity v Bratislave, alebo na pôde iných univerzít v zahraničí. Časť špecifikácia požiadaviek a hrubý návrh riešenia obsahuje informácie o vylepšeniach, ktoré plánuje implementovať tím Jahodoví princovia do hráča tímu Gang Of Six.



Obsah dokumentu

1	Úvod	7
1.1	Znenie zadania	7
1.2	Prehľad dokumentu	8
2	Analýza existujúcich tímov, metód a prostredia RoboCup	9
2.1	Analýza soccer servera	9
2.1.1	Využitie senzorov	9
2.1.2	Módy hry	9
2.1.3	Riadiace príkazy	10
2.1.4	Heterogénny hráči	10
2.1.5	Rozhodca	11
2.2	Analýza existujúcich tímov	11
2.3	Sklo	11
2.3.1	Vnútorný svet hráča	11
2.3.2	Podporné triedy	12
2.3.3	Zvuková komunikácia	12
2.3.4	Taktika	12
2.3.5	Brankár	12
2.3.6	Kouč	12
2.3.7	Záver	12
2.4	Squirrel Squadron	13
2.4.1	Zvuková komunikácia	13
2.4.2	Dynamické zmeny formácií	13
2.4.3	Správanie sa hráča	14
2.4.4	Vizualizácia rozhodovacieho stromu	15
2.4.5	Záver	15
2.5	Dirty Dozen 2001	15
2.5.1	Strategický formalizačný jazyk	15
2.5.2	Kouč	16
2.5.3	Pomocné nástroje	16
2.5.4	Záver	16
2.6	Gang Of Six	16
2.6.1	Analýza hráča tímu UvaTrilearn	17
2.6.2	Zručnosti hráčov a stratégia tímu	19
2.6.3	Zameranie tímu Gang of Six	20
2.6.4	Koordináčne grafy a algoritmus vylučovania premenných	20
2.6.5	Pohľad hráča	22
2.6.6	Prihrávky	23
2.6.7	Driblovanie	24
2.6.8	Brankár	25
2.7	FIITMEDIA	26
2.7.1	Vylepšený algoritmus prihrávania	26
2.7.2	Okolie ohrozeného spoluhráča	27
2.7.3	Driblovanie	27
2.7.4	Komunikácia	27
2.7.5	Záver	27
2.8	Loptoši	27
2.8.1	Brankár	28



2.8.2	Synchronizácia.....	28
2.8.3	Vedúci hry	29
2.8.4	Situácie	30
2.9	Farmári.....	30
2.9.1	Vylepšenia hráča.....	30
2.9.2	Analýza súperových formácií.....	31
2.9.3	Výber heterogénnych hráčov	31
2.9.4	Použitie neurónových sietí na ovládanie nízkych schopností hráča	31
2.9.5	Úprava brankára.....	31
2.9.6	UnitCreator - Tvorba a vyhodnotenie pokrytia subformácií	31
2.9.7	Záver.....	31
3	Hrubý návrh riešenia	32
3.1	Zahrávanie autov	32
3.2	Spôsoby zdokonalenia brankára	33
3.2.1	Chytanie strely smerujúcej popri brankárovi.....	34
3.2.2	Nový spôsob návratu do bránky pri vybehnutí.....	34
3.2.3	Držanie defenzívnej pozície	36
3.2.4	Vylepšenie rozohrávania	37
3.3	Zavedenie a implementácia kouča.....	38
3.3.1	Motivácia.....	38
3.3.2	Úloha kouča.....	39
3.3.3	Rozpoznávanie heterotypov.....	39
3.3.4	Implementácia kouča.....	39
3.4	Heterogénni hráči.....	40
3.4.1	Vlastnosti hráčov v rôznych roliach.....	40
3.4.2	Prečo použiť heterogénnych hráčov?.....	42
3.4.3	Využitie heterogénnych hráčov	42
3.5	Koordináčne grafy a algoritmus eliminácie premenných	42
3.5.1	Úvod.....	42
3.5.2	Koordináčne grafy.....	43
3.5.3	Algoritmus eliminácie premenných	44
3.5.4	Role agentov	45
3.5.5	Stav implementácie rozhodovania pomocou koordináčnych grafov	45
4	Záver.....	47
	Prílohy	48
	Príloha A: Bibliografia	49
	Príloha B: Zoznam tabuliek	50
	Príloha C: Zoznam obrázkov	51



1 Úvod

Témou projektu tímu Jahodoví princovia sú nové stratégie v rámci projektu Robocup. Robocup je svetová iniciatíva v oblasti umelej inteligencie a robotiky. Cieľom tohto projektu je zostaviť do roku 2050 tím robotov, ktorý by dokázal zvíťaziť nad aktuálnymi majstrami sveta vo futbale. Popri tomto dlhodobom ciele je snahou projektu Robocup taktiež nachádzať nové riešenia štandardných problémov z oblastí robotiky a umelej inteligencie, ktoré je potrebné vyriešiť aj v robotickom futbale. Medzi takéto problémy patrí napríklad plánovanie, tímová komunikácia, vnímanie sveta okolo seba či určovanie stratégie.

V rámci iniciatívy Robocup vzniklo prostredie pre simulovanie a sledovanie zápasu dvoch mužstiev. Na báze tohto prostredia sú usporadúvané po celom svete turnaje v robotickom futbale. Súperenie jednotlivých mužstiev a snaha o vytvorenie čo najlepšieho hráča vedú projekt Robocup stále bližšie k naplneniu svojho dlhodobého cieľa. Vďaka predmetu Tvorba softvérového systému v tíme sa môže aj náš tím Jahodoví princovia pridať k tejto celosvetovej iniciatíve.

1.1 Znenie zadania

Téme RoboCup, presnejšie lige simulovaného robotického futbalu sa naši študenti venujú už päť rokov. Tímy študentov, či už v rámci umelej inteligencie alebo tímového projektu, sa snažia vytvárať a vylepšovať programy, ktoré simulujú správanie sa futbalového hráča. Každý tím sa v rámci obmedzení, určených pravidlami hry futbal a špecifikami simulačného prostredia, snaží vytvoriť čo najlepšieho hráča. Mužstvo, vytvorené z takýchto hráčov, by malo vyhrať nad mužstvom súpera. O súťaži a doterajšej činnosti je dosť popísané aj na stránke STU turnaj v simulovanom robotickom futbale (www.fiit.stuba.sk/robocup).

V rámci fakulty sme realizovali viacero súťaží a posledná z nich už bola oficiálnym turnajom iniciatívy RoboCup. Množstvo pozitívnych ohlasov nás priviedlo k vyhláseniu ďalšieho regionálneho turnaja RoboCup v simulovanej lige, opäť na záver akademického roka. Práve množstvo nových prístupov a riešení, ktoré predviedli nielen študenti tímového projektu, ale hlavne študenti umelej inteligencie, nám ukázalo, že možnosti na vylepšovanie hráča nie sú zďaleka vyčerpané a dokonca sa stále rodia prekvapujúce úspešné riešenia. V tomto roku sme preto ako podnázov vybrali – nové stratégie. Znamená to všeobecne hľadanie nových prístupov a stratégií nielen pre hráča, ale aj vo svojej práci, v úpravách zdrojového kódu, podporných aplikáciách, základných aj vyšších schopnostiach hráča, spôsobe učenia a ladenia počas simulácií. Nové stratégie sú komplexnou výzvou do nového kola víťazstiev!

Na upresnenie je vhodné povedať, že v tomto tímovom projekte budeme rozširovať možnosti a vylepšovať správanie sa hráčov, vytvorených vo vlnajších tímových projektoch. Využije sa existujúci zdrojový kód, dokumentácia a aj vytvorené podporné aplikácie. Musí sa tiež zachovať (a podľa možností aj zlepšiť) modularita a tým aj rozširovateľnosť hráča. Zimný semester je vyhradený na oboznámenie sa s celým prostredím, najmä existujúcimi hráčmi a návrhu a prototypovej realizácii jeho vylepšení. Očakáva sa najmä návrh nových prístupov a



stratégií vo všetkých už spomenutých oblastiach. Vybrané prístupy sa overia vytvorením jedného alebo viacerých prototypových rozšírení existujúceho kódu. Dôležitou súčasťou bude vytvorenie plánu implementácie a overovania nových stratégií v nasledovnom semestri. V letnom semestri nás čaká realizácia navrhnutých prístupov a stratégií a ich overovanie. Produkt by mal byť dohotovený v deviatom až desiatom týždni semestra, potom je potrebné venovať sa ladeniu a optimalizácii hráča na súťaž, ktorej výsledky idú do celkového hodnotenia tohto projektu.

1.2 Prehľad dokumentu

V kapitole číslo 2 sa nachádza opis a analýza servera pre simulovanie zápasu dvoch mužstiev. V tejto kapitole sú taktiež uvedené analýzy tímov, ktoré sa venovali problematike Robocupu či už na pôde fakulty, alebo zahraničných tímov. Účelom tejto kapitoly je poskytnúť úvodné informácie o problematike projektu. Zanalyzovaním cudzích tímov sme zistili silné a slabé stránky ich hráčov a získali námety na možné vylepšenia hráča tímu Gang Of Six, v ktorého vývoji sme sa rozhodli pokračovať.

V kapitole číslo 3 sú uvedené vylepšenia, ktoré sa náš tím rozhodol implementovať do hráča tímu Gang Of Six. Zvyčajné ide o vylepšenie tých vlastností hráča Gang Of Six, ktoré náš tím vyhodnotil ako nedostatočné po kvalitatívnej stránke. Pri každom návrhu na vylepšenie sú uvedené dôvody prečo sa tím rozhodol pre danú zmenu. Okrem motivácie je pri jednotlivých návrhoch uvedený aj algoritmus, alebo spôsob akým budú dané návrhy vykonane.



2 Analýza existujúcich tímov, metód a prostredia RoboCup

2.1 Analýza soccer servera

Soccer server je prostredie, ktoré poskytuje virtuálne hracie ihrisko klientom – jednotlivým hráčom. Server simuluje pohyb hráčov, lopty a kontroluje pravidlá hry. Komunikácia medzi serverom a klientami je zabezpečená prostredníctvom protokolu UDP. Za klienta, hráča považujeme program vytvorený v ľubovoľnom programovacom jazyku, ktorý podporuje komunikáciu so serverom pomocou UDP protokolu. Každý klient ovláda jedného hráča na serveri. Aby simulovaný futbalový zápas mohli sledovať aj ľudia bola vyvinutá zobrazovacia súčasť tzv. Soccermonitor, ktorý slúži ako rozhranie zobrazujúce hraciu plochu, hráčov a celú hernú situáciu.

2.1.1 Využitie senzorov

Klienti majú možnosť komunikovať pomocou niekoľkých kanálov, ktoré kvôli lepšej predstavivosti budeme pripodobňovať k ľudským vnemom. Sú to nasledujúce kanály:

- Sluchový kanál (senzor) – hráč využíva tento senzor na prijímanie informácií od spoluhráčov, kouča i od rozhodcu, ktorý je poskytovaný soccer serverom. Počúvanie je realizované funkciou (*hear (Time) (Direction) (Message)*), kde *Time* znamená aktuálny čas, *Direction* určuje smer zdroja a *Message* je samotná správa, ktorá má obmedzenie na jej dĺžku. Maximálna veľkosť môže byť 10 bajtov.
- Zrakový kanál (senzor) – tento senzor ma význam pri vnímaní okolitej situácie. Hráč má možnosť sledovať okolité objekty, ktorá sa nachádzajú v jeho zornom poli. Informácie sú posielané prostredníctvom funkcie (*see (Time) (ObjInfo) (ObjInfo) ...*) v pravidelných intervaloch. Parameter funkcie *see ObjInfo* obsahuje informáciu napr. o natočení, čísle hráča, relatívnej vzdialenosti a mene tímu objektu. Presnosť informácie, ktorú hráč prijíma je závislá od vzdialenosti hráča od sledovaných objektov. Kvalitu pohľadu hráča možno zmeniť použitím funkcie (*change_view (Angel_width) (quality)*). *Angel_width* môže nadobúdať hodnoty *normal*, *wide* a *narrow* a parameter *Quality* nadobúda hodnoty *high* alebo *low*.

Telový (vnemový) senzor - tento senzor umožňuje hráčovi samotnému získať informácie o svojom fyzickom stave využívajúc funkciu (*sense_body (TIME (view_mode QUALITY WIDTH) (stamina ...))*). *Parametre určujú fyzické vlastnosti hráča ako sú energia, natočenie tela, hlavy hráča atď. Informácie sú posielané periodicky.*

2.1.2 Módy hry

Server nám poskytuje niekoľko módov hry, ktoré sú používané počas hry a určujú stav hry.



2.1.2.1 Play_on

Tento hrací mód nastáva po uvedení lopty na hraciu plochu príkazom `kick_off`. Lopta i hráči sú v pohybe.

2.1.2.2 Gól

Ak sa lopta dostane za bránkovú čiaru jednej z bránok, tak nastáva gól. Server preruší zápas na 5 sekúnd, presunie loptu do stredu ihriska a pošle všetkým hráčom informáciu o strelenom góle. Hráči po obdržaní gól majú možnosť zmeniť svoju pozíciu. Ak tak nerobia, tak ich presunie server na náhodnú pozíciu na hracom poli.

2.1.2.3 Presunutie

Nastáva pri rozohrávaní lopty. Simulátor presunie všetkých hráčov, aby boli v prednastavenej minimálnej vzdialenosti 9,15 metra od lopty. Hráči sú presunutý na obvod pomyselnéj kružnice.

2.1.2.4 Polčas

V polčase plánovaného zápasu server preruší hru. Zadefinovaná dĺžka polčasu je 3 000 simulačných cyklov, čo je približne 5 minút. Ak je po ukončení zadefinovaného času skóre nerozhodné, tak sa predlžuje pokiaľ jeden so súperov nestrelí gól.

2.1.2.5 Aut, rok a výkop od brány

Ak lopta opusti hraciu plochu nastáva situácia zvaná aut, resp. aut. Server presunie loptu na pozíciu, ktorú určí podľa zadefinovaných pravidiel.

2.1.3 Riadiace príkazy

Aby sme boli schopný riadiť hráčov, tak máme k dispozícii sadu riadiacich príkazov:

- *Dash* - zrýchlenie hráča v smere natočenia jeho tela.
- *Kick* - kopnutie do lopty s určitým smerom a silou.
- *Catch* - chytenie lopty, možnosť využitia tohto príkazu len brankárom.
- *Say* - posielanie správ svojim spoluhráčom, ktorí sú v dosahu.
- *Otočenie hráča* - zmena natočenia hráčovho tela.
- *Otočenie hlavy* - zmena otočenia hlavy hráča v postavení k telu.
- *Tackle* - pokus o kĺzačku, pri ktorej hráčova snaha je odkopnúť loptu, ktorú vedie súper.
- *Pozornosť na hráča* - zameranie pozornosti na niektorého zo spoluhráčov, správy od tohto hráča sú vypočítané prednostne.

2.1.4 Heterogénny hráči

Socccerserver umožňuje využitie 7 typov hráčov. Na začiatku hry server vygeneruje zo základného typu hráča (typ 0) ďalších 6 typov (typ 1 – 6), ktoré majú niektorú z vlastností



posilnenú (lepšia akcelerácia, silnejší kop a podobne) a zase niektoré vlastnosti degradované. Využitie heterogénnych typov je pre špecifické úlohy hráča ako je napr. obrana.

2.1.4.1 Kouč

Kouč je súčasťou hry a má možnosť poselať informácie svojmu tímu prostredníctvom zvukového kanálu na základe sledovania súpera. Taktiež má možnosť striedania, menenia stratégie hry a podobne. Kouč je v interakcii s hráčmi počas celého priebehu zápasu, ale jeho možnosť komunikácie je obmedzená. Kvôli komunikácii medzi koučom a hráčmi bol vyvinutý programovací jazyk zvaný CLang.

2.1.5 Rozhodca

Rozhodca dohliada na regulárnosť a mení jednotlivé módy hry. Informácie od rozhodcu sú vnímané hráčom pomocou sluchového senzoru. Tieto správy hráč počuje vždy, a to aj v prípade, že mu bola poslaná správa od iného hráča, ktorú počuje takisto.

2.2 Analýza existujúcich tímov

Predpokladom pre realizáciu projektu je využitie už naimplementovanej funkčnosti z iných tímov. Nasleduje analýza vybraných domácich a zahraničných tímov. Účelom analýzy je sumarizácia metód používaných na modifikáciu správania hráča.

2.3 Sklo

Tím Sklo bol vytvorený v školskom roku 2003/2004 na pôde FIIT STU v rámci predmetu Tímový projekt. Vychádzal z práce tímu Stjupid Dox, ktorá vznikla podobným spôsobom rok predtým.

Hlavným zameraním sa tímu Sklo bolo:

- rozšírenie a prepracovanie vnútorného sveta hráča
- sledovaniu situácie na ihrisku
- komunikácie hráčov
- rozohrávka, výkop
- rozlišovanie súperových heterotypov koučom
- správanie sa hráčov na rôznych pozíciách

2.3.1 Vnútorný svet hráča

Autori upravili vnútorný svet hráča, odstránili dátovú duplicitu, kde sa ukladali informácie v zdrojovom kóde dvakrát, samostatne pre spoluhráčov a samostatne pre súperov. Pridali taktiež funkcie na nájdenie hráča podľa rôznych kritérií. Hráč dokáže predikovať počet simulačných krokov, za ktoré dokáže dobehnúť hráč k určitému bodu alebo taktiež dobehnúť pohybujúci sa predmet (loptu).



2.3.2 Podporné triedy

Pre vývoj sofistikovanejších algoritmov a uľahčenie vývoja a ladenia, si tím Sklo vytvorili pomocné triedy, ktoré im pomáhali pracovať s objektmi typu úsečka, kružnica, priamka, polpriamka a taktiež triedu na logovanie.

2.3.3 Zvuková komunikácia

Správy sa delia na dva druhy. Na bežné správy a urgentné. Bežné správy hráč kričí len vtedy ak je na rade. V týchto správach hráč kričí svoj vnútorný svet. Ostatné správy sú urgentné a kričia sa aj vtedy, keď hráč nie je na rade.

2.3.4 Taktika

Tím sa snažil implementovať dve špeciálne taktické funkcie. Prvá funkcia slúži na dynamické určovanie cieľa, ktorým smerom bude driblovať hráč s loptou. Druhým algoritmom je výber miesta na ktoré má hráč kopnúť loptu, keď nemôže ďalej bezpečne driblovať.

Taktika samotných hráčov sa odlišuje podľa toho na akej pozícii hrá daný hráč. Hráči sa delia na útočníkov, stredopoliarov, obrancov a brankára. Navyše sa rozlišuje či hráč má alebo nemá loptu. Hráči sa v rámci svojho pohybu po ihrisku snažia čo najviac šetriť energiou. Majú vytvorené domovské pozície na ihrisku, na ktoré sa snažia vrátiť ak nemajú loptu alebo tá nie je blízko nich.

2.3.5 Brankár

Brankár sa pred výkopom snaží zhodnotiť situáciu na ihrisku. Na základe zozbieraných údajov sa presunie na také miesto, z ktorého je najvýhodnejšie kopat loptu.

2.3.6 Kouč

Pre právne rozhodovanie hráčov je potrebné vedieť predpokladať ako sa zachová súper. Túto informáciu je však pomerne ťažké získať pre hráča, preto sa využíva na túto funkcionálnu kouč. Jeho úlohou je zistiť súperove heterotypy a odovzdať správu o nich hráčom.

Šum v hre spôsoboval, že kouč dokázal určiť iba ich rýchlosť a zrýchlenie heterotypov, nedokázal presne určiť heterotypy podľa otáčania hráča a vzdialenosti z ktorej hráč dokázal kopnúť do lopty.

2.3.7 Záver

Tím Sklo má pomerne silnú obranu a brankára, vďaka čomu dostávajú pomerne málo gólov. Vzhľadom na to že majú prehustenú obranu (5 hráčov), ty chýbajú v útoku, čo vedie k malému počtu strelených gólov.

Hoci je hráč vyvinutý pomerne dávno (2004) a nie je veľký predpoklad že sa budú používať jeho zdrojové súbory ďalej, niektoré myšlienky a vlastnosti by sa mohli použiť i pri návrhu a implementácii nami vyvíjaného hráča.



2.4 Squirrel Squadron

Squirrel Squadron je tím zo šk. rok 2004/2005. Vyvinutý hráč je postavený na hráčovi tímu Sklo (z roku 2003/2004), ktorého vylepšuje a opravuje niekoľko jeho chýb. Na turnaji RoboCup at FIIT'05 sa tím Squirrel Squadron umiestnil na poslednom mieste v základnej trojčlennej skupine B s jednou remízou (skóre 0:0) a jednou prehrou (skóre 0:3).

Tím Squirrel Squadron sa zamerlal na riešenie problémov nasledujúcich oblastí:

- zvuková komunikácia hráčov
- dynamické zmeny formácií
- správanie sa hráča

2.4.1 Zvuková komunikácia

Hráči si môžu posielat' zvukové správy, ktorými sa navzájom informujú o stave na ihrisku, ktorý informovaný spoluhráč vďaka svojej pozícii alebo uhlu pohľadu apod. nevníma alebo o svojich úmysloch. Tím implementoval alebo prevzal z predchádzajúceho tímu správy za sebou, prihraj, chytaj, mám, bež, pusti, správy na riadenie ofsajdovej pasce, pokry, AttentionTo a správy na komunikáciu s koučom. Podrobnejšie sa tím zamerlal na správy pokry a pusti.

2.4.1.1 Správa „pokry“

Slúži na zabránenie krížnych prihrávk pred bránou, z ktorých často padajú góly. Túto správu vysielal brankár, ktorý má najlepší prehľad o dianí pred bránou svojmu voľnému spoluhráčovi, aby zabezpečil pokrytie najbližšieho voľného protihráča.

2.4.1.2 Správa „pusti“

Slúži na to, aby za loptou nebežalo súčasne viacero spoluhráčov, ale len ten najbližší. Správu kričí hráč, ktorý si myslí, že je najbližšie k lopte. Ostatní hráči si po prijatí správy pusti nastavili príznak „nebež za loptou“.

2.4.2 Dynamické zmeny formácií

Tím Squirrel Squadron implementoval dynamické zmeny formácií na základe stavu skóre alebo pokročilosti času v zápase, príp. na základe vyhodnotenia formácie súpera. Pre tento účel bolo ihrisko logicky rozdelené na štvorcové zóny. Tím preddefinoval niekoľko štandardných formácií a pridal možnosť dynamicky vytvoriť novú formáciu. Pre potrebu hry vo formácii bola hráčom definovaná ich domovská pozícia.

2.4.2.1 Využitie štatistík

Povely o zmene formácie vysielal kouč, ktorý pri svojom rozhodovaní využíva nasledujúce štatistiky:

- výskyt lopty v určitých zónach ihriska



- poloha spoluhráčov, resp. protihráčov, v zónach
- kontrolovanie lopty

2.4.2.2 Použité algoritmy

Na vyhodnocovanie štatistík boli použité nasledujúce 3 algoritmy:

- záplavový algoritmus
- výškový algoritmus
- algoritmus lokálnych maxím

Zo spomínaných algoritmov prvé dva dosiahli dobré výsledky, tretí sa ukázal ako nepoužiteľný.

2.4.2.3 Zmena formácie

Kouč prehodnocuje aktuálnu formáciu pri jednej z nasledujúcich podmienok:

- Ubehlo 1500 cyklov od posledného nulovania štatistík.
- Tím dostal gól.

Primárnym rozhodovacím kritériom je rozdiel medzi inkasovanými a strelenými gólmi. Ak je rozdiel priveľký, kouč volí obrannú stratégiu (lebo vie, že „nemá šancu“). Ak je rozdiel malý, volí sa mierne útočná stratégia (snaha o vyrovnanie). Ak je rozdiel záporný, t.j. tím vyhráva, volí sa opäť obranná stratégia.

Ďalšími kritériami sú percentuálny výskyt lopty v obrannom pásme a počet vlastných obrancov.

Pri zmene formácie sa štatistiky nulujú.

2.4.3 Správanie sa hráča

Správanie sa hráča je ovplyvnené najmä nasledujúcimi faktormi:

- vzdialenosť hráča od lopty
- rola hráča (obranca, stredopoliar, útočník – každý má iné priority)
- postavenie spoluhráčov
- postavenie protihráčov
- stav skóre
- zóna, v ktorej sa hráč nachádza (útok, stred, obrana, krídla)

2.4.3.1 Zóny

Správanie hráčov sa mení podľa toho, v akej fáze je hra a v akej zóne sa nachádzajú. V obrannej zóne sa snažia obrancovia dostať loptu „do bezpečia“ (nahrávkou ďalej od brány



alebo odkopom mimo ihriska). V obrannej fáze hry sa stredopoliari snažia pokrývať nabiehajúcich súperových hráčov a útočníci sa snažia nebyť v ofsajde.

Pri hre v strednom pásme sú hráči takticky rozostavení v príslušných zónach.

V útočnej fáze hry sa snažia obrancovia vystaviť súperových útočníkov do ofsajdovej pasce a nedovoliť im získať prihrávku. Stredopoliari očakávajú oslobodzujúce odkopy súperových obrancov a vracajú loptu späť do šestnástky. Útočníci okrem „ťahu na bránu“ rozvíjajú hru po krídlach.

Pri implementácii zón narazil tím na problém pri pohybe hráča na rozhraní zón: Hráč pohybujúci sa na rozhraní zón rovnobežne s týmto rozhraním často mení svoje „úmysly“, pretože v susedných zónach vyhodnotí situáciu odlišne. Tím riešil problém implementovaním dynamických hraníc zón.

2.4.4 Vizualizácia rozhodovacieho stromu

Tím Squirrel Squadron vizualizoval rozhodovacie stromy hráča tímu Sklo, z ktorého vychádzal. Vizualizáciou dosiahol lepšie pochopenie správania sa hráča a odhalil viacero chýb. Výsledkom bolo viacero vylepšení:

- zlepšená strelba na bránu
- vylepšený dribling
- zníženie rizika straty lopty v strednej časti hry skrátením driblingu záložníkov.

2.4.5 Záver

Určite aj vzhľadom k tomu, že tímu sa nepodarilo zrealizovať všetky vylepšenia, skončil tím na poslednom mieste v turnaji RoboCup at FIIT'05. Keďže je to už starší hráč (pracovali na ňom už 4 tímy), nie je podľa môjho názoru výhodné pokračovať v jeho ďalšom vývoji. Obsahuje však niekoľko zaujímavých vlastností, ktoré by sa možno oplatilo prebrať do nášho hráča.

2.5 Dirty Dozen 2001

Tím ORCA (Osnabrueck RoboCup Agents) vznikol v roku 2000 na univerzite v Osnabruecku v Nemecku. Jeho hráč, Dirty Dozen, bol postavený na kóde CMU (Carnegie Mellon University), ktorý bol zverejnený v roku 1999.

Stratégia hráča je postavená na rozhodovacom strome. Informácie prijaté zo servera sú spracované a uložené do vnútornej reprezentácie sveta. Na základe stavu reprezentácie sveta a rozhodovacieho stromu sa rozhodne o nasledujúcej akcii. Rozhodovací strom je reprezentovaný ako množina if-then konštrukcií.

2.5.1 Strategický formalizačný jazyk

S vývojom štandardizovaného jazyka koučov Clang, diskutoval tím o tom, ako integrovať tieto informácie do svojho hráča. Prišli s nápadom použiť Strategy Formalization Language



(SFL), čo je rozšírenie jazyka Clang. Hráč ma teda dva zdroje informácií pre rozhodovanie. Je to vnútorný model sveta a tiež príležitostné informácie od kouča, ktoré si hráč ukladá do bázy znalostí. Tá je už dopredu naplnená vlastnými strategickými informáciami. Dá sa povedať, že informácie od kouča bázu znalostí len rozširujú a majú za úlohu zlepšiť hru podľa dopredu známych stratégií.

SFL systém implementovali ako tri komponenty: Matcher, Selector a Effector. Najdôležitejšou časťou je Matcher, ktorý porovnáva znalosti v báze s informáciou z vnútorného modelu sveta. Na základe tohto porovnania vyberie Selector najvhodnejšie pravidlo použitím určitej heuristiky. Vybrané pravidlo následne zrealizuje Effector.

2.5.2 Kouč

Keďže tím ORCA zimplementoval do svojho hráča reaktivitu na informácie od kouča podľa štandardu Clang, mohol experimentovať s rôznymi dostupnými koučami. Nakoniec však vytvorili vlastného, ktorý sa zameriaval na poskytovanie tých informácií, ktoré boli najvhodnejšie pre nimi vytvorené stratégie v hráčovi.

2.5.3 Pomocné nástroje

Na tomto hráčovi sú zaujímavé nástroje, ktoré tím pri vývoji svojho hráča použil. Tieto nástroje by sme mohli použiť aj my pri vývoji nášho hráča.

Logfile Analyzer – číta logy a reprezentuje ich v čitateľnejšej a zrozumiteľnejšej podobe.

The Gauntlet – nástroj pre automatizované porovnávanie kvality tímov. Tento program spustí sériu zápasov medzi aktuálnym hráčom a niektorými dostupnými hráčmi. Výstup z tohoto turnaja je reprezentovaný v podobe html stránok.

2.5.4 Záver

Hráč Dirty Dozen je dosť starý hráč (z roku 2001), preto by som neodporučil vybrať si práve jeho a rozširovať ho. Má však niekoľko zaujímavých črt, ktoré som popísal vyššie. Nimi by sme sa mohli inšpirovať. Určite by sme však mohli použiť pomocné nástroje, ktoré využil tento tím.

2.6 Gang Of Six

Tím Gang Of Six bol založený v roku 2006 na pôde Slovenskej technickej univerzity v Bratislave a svoju činnosť ukončil v roku 2007. Tím bol tvorený šiestimi študentmi. Hráč vytvorený tímom Gang Of Six dosahoval vynikajúce výsledky a zvíťazil na záverečnom turnaji organizovanom na pôde Fakulty informatiky a informačných technológií.

Tím Gang Of Six pokračoval vo vývoji hráča tímu UvaTrilearn, ktorý však poskytol len neúplnú verziu svojho hráča. Časť verzie, na ktorej pracoval tím Gang Of Six bola vytvorená tímom FCPortugal. Aj napriek tomu, že tím UvaTrilearn neposkytol kompletnú verziu svojho hráča bola takto získaná funkcionálna na veľmi dobrej úrovni. Zdrojové kódy hráča vytvoreného tímom UvaTrilearn boli napísané prehľadne a dobre zdokumentované. V trende

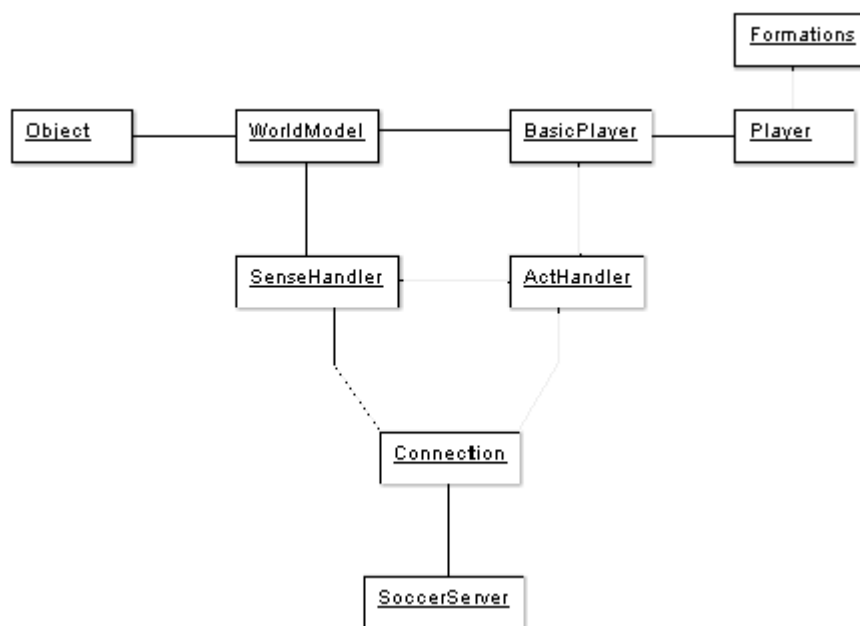


prehľadného a dobre zdokumentovaného kódu pokračoval pri svojom vývoji aj tím Gang Of Six.

V nasledovnej podkapitole opíšeme základné princípy hráča tímu UvaTrilearn, keďže túto pôvodnú funkcionality zachoval aj tím Gang Of Six. Vylepšenia tímu Gang Of Six sú opísane v kapitole.

2.6.1 Analýza hráča tímu UvaTrilearn

Základnou myšlienkou hráča tímu UvaTrilearn je rozlíšenie 3 hlavných úloh – vnímanie, rozhodovanie, konanie. Z tohto rozdelenia vyplýva aj architektúra hráča, znázornená na obr. č. 1.



Obr. 1: Architektúra hráča tímu UvaTrilearn.

Na najnižšej úrovni pracuje trieda Connection, ktorá zabezpečuje spojenie so soccer serverom. Spojenie je vykonávané cez UDP socket. Po nadviazaní spojenia so serverom začínajú prichádzať správy. Získavanie a prvotná analýza týchto správ je úlohou triedy SenseHandler. Na posielanie správ serveru bola vytvorená trieda ActHandler. Táto trieda obsahuje rad správ, ktoré sa majú vykonať v danom takte, keďže v jednom takte je možné vykonať viacej akcií naraz. Niektoré akcie však môžu byť vykonané len raz v jednom takte (bež, otoč sa, kopni...). Poslané správy sa synchronizujú aj s triedou WorldModel, tak aby mala vždy aktuálny prehľad o stave na ihrisku.

2.6.1.1 Trieda WorldModel

Trieda WorldModel je opísaná v samostatnej kapitole, keďže predstavuje najdôležitejšiu a najzložitejšiu triedu v architektúre hráča tímu UvaTrilearn. Model sveta obsahuje informácie o aktuálnom stave zápasu a okolia. Model sveta zároveň predstavuje pravdepodobnostnú reprezentáciu vytvorenú na základe predošlých vnemov.



Jednotlivé informácie o aktuálnom stave zápasu a okolia ako aj pravdepodobnostné odhady budúceho stavu na ihrisku môžu hráči získavať cez tzv. prístupové metódy („setters“). Okrem samotných informácií odosiela model sveta aj časovú pečiatku, ktorá obsahuje informáciu o tom, kedy bola správa odoslaná. Hráč je na základe tejto informácie schopný rozlišovať medzi aktuálnymi a neaktuálnymi informáciami.

Pri spracovávaní správ odoslaných soccer serverom je potrebné uvedomiť si, že server posiela informácie o pozíciách vzhľadom na pozíciu hráča. Tieto pozície sú teda relatívne. Hráč si ich musí po prijatí previesť do tzv. globálnej formy. Na túto transformáciu mu môžu poslužiť informácie o umiestnení stacionárnych objektov (čiary, brána...).

Model sveta obsahuje množstvo užitočných atribútov a metód, ktoré sú potrebné pre hráčov pri určovaní ich nasledovných akcií. Z tohto dôvodu v nasledujúcej časti opisujeme základné skupiny atribútov a metód, ktoré sa nachádzajú v modeli sveta.

Atribúty:

Informácie o okolí: Tieto informácie sú špecifické pre prostredie poskytované serverom. Medzi informácie o okolí patria napríklad parametre servera a informácie potrebné pre nastavenie heterotypov hráčov.

- Informácie o zápase: Táto skupina obsahuje základné informácie o zápase. Patrí sem napríklad informácia o čase, ktorý uplynul od začiatku zápasu či číslo konkrétneho hráča. Okrem toho obsahuje aj informácie o strane, z ktorej tím hrá, alebo aktuálny gólový rozdiel.
- Informácie o objektoch: Objekty na ihrisku sa delia na dynamické a stacionárne. Medzi dynamické objekty patria objekty, ktoré sa pohybujú (lopta, hráči). Stacionárnymi objektmi sú napríklad čiara, brána, alebo rohové zástavky.
- Informácie o akciách: Túto triedu informácií tvorí zoznam akcií, ktoré vykonali jednotliví hráči v minulosti. Pomocou týchto informácií je možné zistiť či bola daná akcia vykonaná, alebo koľko akcií daného typu bolo vykonaných.

Metódy:

Získavacie metódy: Sú to tzv. “getters” metódy. Slúžia na získavanie informácií z modelu sveta. Názvy týchto metód sú v tvare `getMenoAtribútu`.

- Aktualizačné metódy: Slúžia na aktualizáciu informácií v modeli sveta. Aktualizačné metódy majú názvy v tvare `setMenoAtribútu`.
- Predpovedajúce metódy: Sú to nízkoúrovňové metódy predpovedajúce vývoj jednotlivých parametrov v budúcnosti. Predpovedanie budúceho stavu prebieha na základe informácií získaných z vnemov. Medzi tieto metódy patria metódy určujúce stav hráča po vykonaní určitej akcie, metódy predpovedajúce stav ostatných pohybujúcich sa objektov na ihrisku (tento typ predpovedí je však veľmi náročný)



a metóda predpovedajúca čas, ktorý potrebuje hráč na presun k určenej pozícii na ihrisku.

- Vysokoúrovňové metódy: Model sveta dokáže na základe informácií získaných z vnemov odvodiť aj tzv. vysokoúrovňové informácie. Sú to napríklad metódy, ktoré vrátia počet hráčov v určitej oblasti, metódy, ktoré vracajú najrýchlejšieho hráča vzhľadom na nejakú pozíciu, alebo metóda, ktorá určuje smer najbezpečnejšej trajektórie lopty medzi protihráčmi.

2.6.2 Zručnosti hráčov a stratégia tímu

Triedy BasicPlayer a Player obsahujú informácie potrebné pre vykonanie akcií. Zručnosti hráčov rozdelil tím UvaTrilearn do troch kategórií: zručnosti nízkej úrovne, zručnosti strednej úrovne a zručnosti vyššej úrovne. Jednotlivé kategórie zručnosti sú opísané v nasledujúcej časti dokumentu.

2.6.2.1 Zručnosti nízkej úrovne:

Zručnosti na tejto úrovni je schopný priamo vykonať aj server. Zvyčajne ide o jednoduché povely, ktoré neobsahujú žiadne zložité rozhodovacie procesy. Medzi zručnosti nízkej úrovne patria okrem iných aj nasledovné zručnosti:

- Otočenie tela hráča k zvolenému bodu na ihrisku
- Hľadanie lopty v blízkosti hráča
- Zachytenie lopty
- Zakričanie

2.6.2.2 Zručnosti strednej úrovne:

Zručnosti na tejto úrovni sú vytvorené zo zručností nízkej úrovne a môžu už obsahovať jednoduché rozhodovacie procesy. K jednotlivým zručnostiam na strednej úrovni patria napríklad:

- Odkopnutie lopty do určitého priestoru tak, aby v cieľovom priestore mala požadovanú rýchlosť.
- Pohybovanie sa po zvolenej strane.
- Zastavenie lopty ak sa nachádza blízko lopty.
- Otočenie sa tela hráča spolu s loptou.

2.6.2.3 Zručnosti vysokej úrovne:

Táto skupina zručnosti pozostáva zo zručnosti strednej úrovne a obsahujú v sebe zložitú rozhodovaciú logiku. Medzi tieto zručnosti, ktoré predstavujú najvyššiu úroveň abstrakcie z pohľadu zručnosti patria napríklad:



- Driblovanie
- Priama prihrávka lopty objektu
- Bránenie bránkovej čiary
- Predkop lopty vybranému objektu tak, aby ju dobehol v behu

Stratégia tímu je definovaná ako kolektívny plán, ktorý je vykonávaný za účelom dosiahnutia spoločného cieľa.

Stratégiu tímu ovplyvňuje najmä :

- Sila súperovho tímu
- Štýl súperovej hry
- Stav hry
- Aktuálna herná situácia
- Dostupnosť zdrojov

Formáciu v tíme UvaTrilearn dodržia najmä pasívny hráči (hráči, ktorí sú ďaleko od lopty). Ak sa hráč vzdialil od lopty na určitú vzdialenosť jeho úlohou je vrátiť sa na miesto, ktoré mu je priradené formáciou tímu. Táto časť hráča bola prebratá od tímu FC Portugal. Tím Gang Of Six formácie tímu nemenil, keďže tie sú vytvorené na vynikajúcej úrovni a spĺňali všetky požiadavky tímu Gang Of Six.

2.6.3 Zameranie tímu Gang of Six

Tím Gang Of Six zachoval funkcionality tímu UvaTrilearn opísanú v predchádzajúcich kapitolách a zameril sa na prepracovanie a zlepšenie nasledovných oblastí:

- Koordinačné grafy a algoritmus vylučovania premenných (náhrada za komunikáciu hráčov)
- Pohľad hráča
- Prihrávky
- Driblovanie
- Brankár

Jednotlivé oblasti, na ktorých pracoval tím Gang of Six, sú bližšie opísané v nasledujúcich kapitolách.

2.6.4 Koordinačné grafy a algoritmus vylučovania premenných

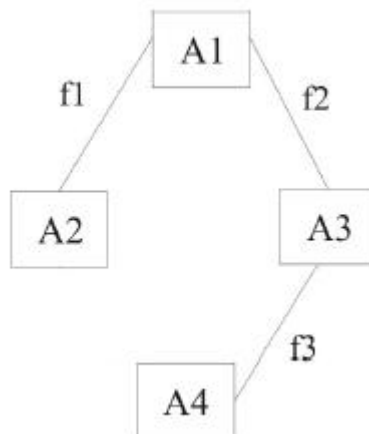
Metódu rozhodovania pomocou koordinačných grafov použil tím Gang of Six ako náhradu za neexistujúcu vzájomnú komunikáciu hráčov. Koordinačné grafy sú známou metódou používanou na vysokoúrovňové rozhodovanie v kooperatívnych multiagentových systémoch.



Koordinácia je proces, v rámci ktorého si jednotliví hráči vyberú akcie takým spôsobom, že tento výber vyústí do optimálnej akcie celého mužstva. Celý proces pozostáva z dvoch krokov. V prvom kroku sa vždy pri zmene situácie na ihrisku vytvorí koordinačný graf, v ktorom uzly predstavujú hráčov a tí hráči, ktorých akcia je navzájom závislá, sú prepojení hranou. Po vytvorení koordinačného grafu sa na vyriešenie koordinačného problému použije algoritmus vylučovania premenných.

Koordinačný problém je možné teoreticky opísať tak, že každý hráč ma v nejakom okamihu k dispozícii množinu akcií a výnosovú funkciu. Spoločná akcia celého tímu je definovaná ako karteziánsky súčin akcií jednotlivých množín akcií všetkých hráčov v tíme. Koordináčnym problémom v tejto situácii je nájsť také akcie jednotlivých hráčov, aby bola výnosová funkcia spoločnej akcie celého tímu čo najvyššia.

Prvým krokom pri riešení koordinačného problému je vytvorenie koordinačného grafu reprezentujúceho aktuálny stav na ihrisku. Koordináčny graf je najrozšírenejším spôsobom reprezentácie koordinačného problému. Uzly grafu predstavujú jednotlivých hráčov tímu a hrany medzi uzlami reprezentujú závislosť medzi akciami daných hráčov. Príklad koordinačného grafu je znázornený na obr. 2.



Obr. 2: Príklad koordinačného grafu.

Na obrázku je znázornený koordinačný graf pre koordinačný problém štyroch hráčov. Hráčom A1,A2,A3,A4 zodpovedajú príslušné uzly v grafe. Ako vyplýva z obrázku hráč A1 musí koordinovať svoju akciu s hráčmi A2 a A3. Hráč A2 musí koordinovať svoju akciu s hráčom A1, hráč A3 s hráčmi A1 a A4 a nakoniec hráč A4 musí koordinovať svoju akciu s hráčom A3. Hrany f1, f2 a f3 zároveň reprezentujú aj výnosovú funkciu pre danú dvojicu hráčov. Funkcia f1 tak napríklad reprezentuje užitočnosť zvolenej dvojice akcií pre hráčov A1 a A2.

Druhým krokom v procese hľadania optimálnej akcie tímu je algoritmus vylučovania premenných. Tento algoritmus slúži na vyriešenie lokálnych koordinačných problémov. Podstatou tohto algoritmu je, že sa pre jedného hráča určí lokálny maximalizačný krok



a následne sa tento hráč z koordinačného grafu vylúči. Tento proces sa opakuje až kým sa z grafu nevyľúčia všetci hráči. Tímu Gang Of Six sa podarilo implementovať len časť určenia koordinačného problému. Algoritmus vylučovania premenných zostal nedokončený a úlohou nášho tímu je implementácia tohto algoritmu. Z toho dôvodu je bližší opis algoritmu uvedený v časti venujúcej sa návrhu riešenia tohto algoritmu.

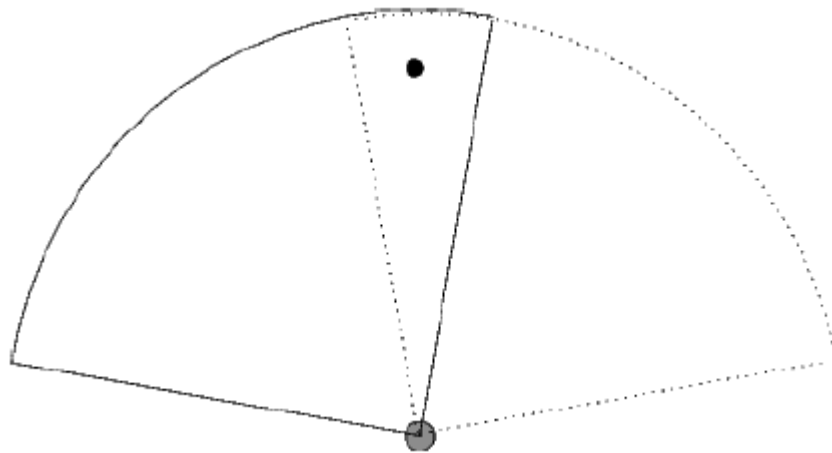
Na záver časti o koordinačných grafoch je ešte potrebné dodať, že tím Gang Of Six nevytváral graf z hráčov, ale z ich rolí. Tento krok bol potrebný z výkonnostných dôvodov algoritmu, keďže vytvárať graf zo všetkých hráčov by bolo neúmerné náročne na výpočtové prostriedky. Jednotlivým hráčom sa tak prideli jedna zo štyroch rolí a celý koordinačný problém sa rieši len pre tieto štyri role. Rola sa hráčom určuje na základe ich vzdialenosti od lopty. Hráčom tímu Gang Of Six môže byť priradená jedna z nasledovných rolí:

- Active – očakáva sa, že hráč vykoná akciu s loptou
 - passer – hráč môže kopnúť do lopty
 - interceptor – hráč nemôže kopnúť do lopty
- Receiver – očakáva sa, že hráč dostane prihrávku
- Passive – hráč sa presúva na svoju strategickú pozíciu
- Brankár – defaultne priradená rola.

2.6.5 Pohľad hráča

Pôvodný hráč tímu UvaTrilearn pozerá na loptu pod uhlom 90 stupňov. Tým, že hráč používal len jeden spôsob pozerania na loptu, mal k dispozícii len obmedzený prehľad o dianí na ihrisku. Hráč tak nevedel, čo sa deje vedľa neho a za ním. Tím Gang of Six si dal za úlohu zlepšiť pohľad hráča tak, aby bolo možné upravovať veľkosť zorného uhla hráča podľa potreby. Neustálym otáčaním hlavy tak hráč získava čo najlepší prehľad o dianí na ihrisku a zároveň nestráca z dohľadu ani loptu. Hráčovi tímu Gang Of Six je možné priradiť jeden z troch zorných uhlov (45,90,180) na základe vzdialenosti hráča od lopty. Čím je hráč bližšie pri lopte tým je mu priradený menší zorný uhol. Ak je hráčovi priradený väčší zorný uhol tak logicky vidí väčšiu časť ihriska, ale na druhej ak je hráčovi priradení menší zorný uhol tak dostáva vizuálny vnem o dianí na ihrisku častejšie.

Okrem možnosti zvolenia zorného uhlu pohľadu, tím Gang Of Six zaviedol u hráčov aj neustále otáčanie hlavou doľava a doprava tak, aby získal ešte lepší prehľad o hre. Situácia zobrazujúca funkciu otáčania hlavou je znázornená na obrázku číslo 3.



Obr. 3: Príklad otáčania hlavou

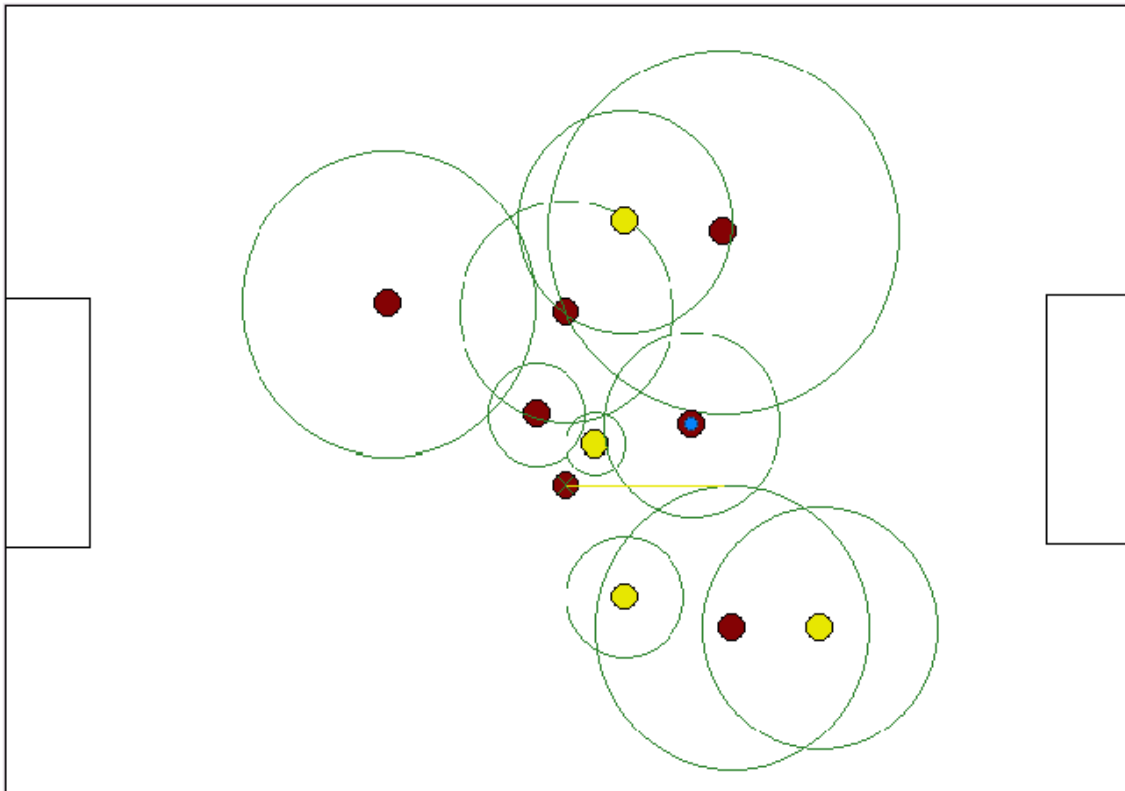
2.6.6 Prihrávky

Tím Gang Of Six sa zamerával aj na zlepšenie prihrávok hráča. Tím vytvoril postupne až štyri verzie prihrávok. Prvú verziu prihrávok tím zamietol, kvôli tomu, že pri prihrávaní sa nebrali do úvahy súperovi hráči a navyše prihrávky boli pre hráčov často nedostihnuteľne.

V druhej verzii prihrávok tím Gang Of Six vylepšil presnosť prihrávok, ale stále nevyriešil problém neuvažovania súperových hráčov pri prihrávkach.

Tento problém tím Gang Of Six vyriešil až v tretej verzii prihrávok. Táto verzia bola zavrhnutá kvôli tomu, že hráči sa snažili prihrávať do oblastí, kde sa nachádzalo najviac spoluhráčov. Pomerne často sa tak stávalo, že útočníci prihrávali loptu stredopoliarom a tí ju zase prihrávali obrancom. Taktiež nastávala situácia, že hráč prihrával priamo súperovi, ak sa pri tomto súperovi vyskytovalo viacero spoluhráčov.

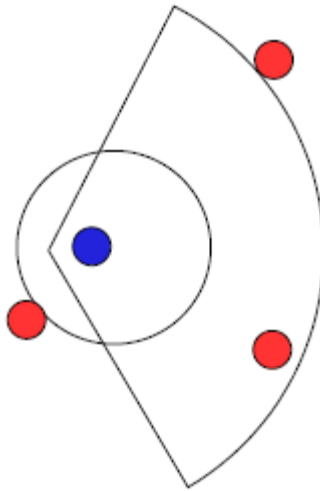
Tím Gang Of Six pri zisťovaní hráča, ktorému by mala byť adresovaná prihrávka, postupuje tak, že okolo každého hráča vytvorí kruh. Vnútro kruhu predstavuje priestor, v ktorom je hráč schopný prihrávku spracovať. Každému hráčovi sa priradí hodnota podľa polomeru jeho osobného kruhu. Čím je polomer kruhu väčší tým je väčšia pravdepodobnosť, že hráč prihrávku spracuje. V poslednej štvrtej verzii tím Gang Of Six začal zvýhodňovať prihrávky smerom dopredu, ale zachoval aj podmienky z predchádzajúcich verzií prihrávok. Na rozdiel od tretej verzie prihrávok navyše prihrávajúci hráč už neuprednostňuje priestor, kde je koncentrovaných čo najviac spoluhráčov, ale konkrétneho spoluhráča. Príklad rozhodovania sa prihrávajúceho hráča a osobných kruhov ostatných hráčov je znázornený na obrázku číslo 4.



Obr. 4: Príklad osobných kruhov hráčov

2.6.7 Driblovanie

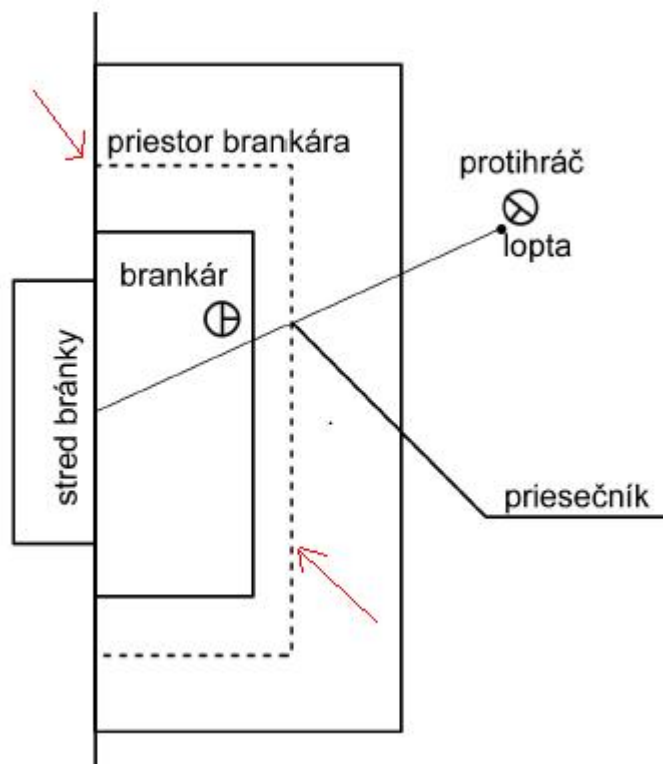
Tímu Gang Of Six sa podarilo taktiež zlepšiť driblovanie hráča. V pôvodnom hráčovi tímu UvaTrilearn existoval len jeden typ driblovania. Tím Gang Of Six implementoval 3 typy driblovania, medzi ktorými sa hráč môže rozhodovať. Medzi nove funkcie patrí aj metóda `isFreeCone()`, ktorá sa používa na určenie, či sa v hráčovom okolí nachádza nejaká prekážka. Funkcia berie ako parametre smer hráča a polomer kruhového výseku. Jej výstupom je hodnota `true`, alebo `false`, ktorou funkcia buď odporučí driblovanie, alebo ho zamietne. Funkcia ako prvé vypočíta polohu kruhu, ktorý sa nachádza pred hráčom. Ak sa v tomto kruhu nachádza protihráč funkcia automaticky vracia `false`. Inak zistí či sa nenachádza súper v kruhovom výseku pred hráčom. Teraz taktiež vracia `false`, ak sa v tomto kruhovom výseku nachádza protihráč, inak vracia `true`. Opísaná situácia je znázornená na obr. č. 5.



Obr. 5: Príklad rozhodovania sa pri driblovaní.

2.6.8 Brankár

Tím Gang Of Six vylepšil aj správanie sa brankára, keďže nebol spokojný zo správaním sa pôvodného brankára vytvoreného tímom UvaTrilearn. V tejto pôvodnej verzii brankár nemohol prekročiť zónu medzi bránkoviskom a jedenástkou znázornenú na obr. č. 6.



Obr. 6: Povolená zóna brankára.

Tím Gang Of Six umožnil brankárovi vybehnúť aj za túto zónu za určitých podmienok. Na tomto mieste je však potrebné podotknúť, že v mnohých prípadoch ak brankár vybehol za túto zónu, sa nestihol načas vrátiť naspäť do brány.

2.7 FIITMEDIA

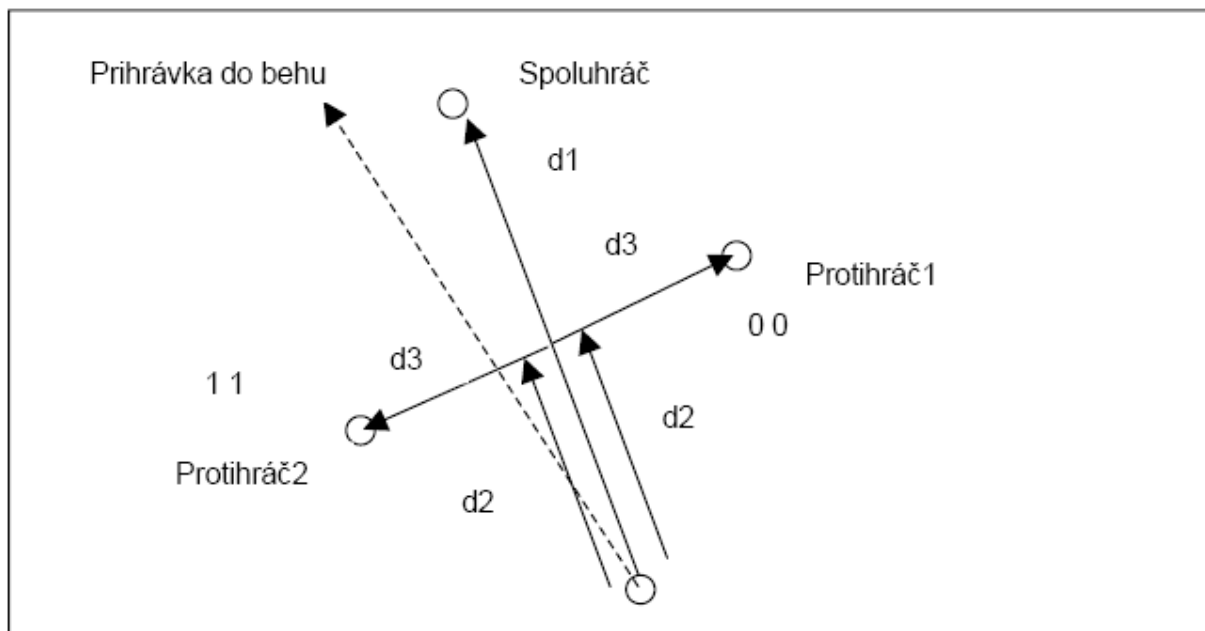
Tím FIITMEDIA je víťazný tím turnaja RoboCup, ktorý sa konal na Fakulte informatiky a informačných technológií (FIIT) STU v tomto roku (teda v roku 2006). Tvorcovia nadväzovali na tím FC Farmári z predošlého roka, ktorý podobne zvíťazil na turnaji RoboCup na FIIT v roku 2005.

Tím mal veľké plány, ale väčšinu z nich sa im nepodarilo uskutočniť lebo väčšinu času venovali opravovaniu kódov a nedostatkov, ktoré sa nachádzali v prevzatom hračovi.

2.7.1 Vylepšený algoritmus prihrávania

Tím FC Farmári obsahuje neurónovú sieť. ňou ohodnocujú prihrávky spoluhráčom hodnotou pravdepodobnosti úspechu. Na základe toho sa vyberie jedná z najpravdepodobnejších, s najväčším možným úspechom.

FIITMEDIA zmenila pôvodný systém prihrávania, ktorý vždy vykonával prihrávky do behu a zaviedla delenie prihrávok na prihrávky do behu a prihrávky na miesto. Takúto prihrávku do behu ilustruje **Error! Reference source not found.**č. 7.



Obr. 7: Hráč v spodnej časti prihráva spoluhráčovi do behu, pričom prihrávku ohrozuje protihráča 2.

Bolo potrebné upraviť pôvodné vstupné vektory neurónovej siete, ktorá ohodnocuje prihrávky pravdepodobnosťami úspešnosti. Prihrávky do behu boli teda pridané priamo do neurónovej siete, ktorá teraz môže odporučiť i takýto druh prihrávky.



Tím študoval možnosti nahrávania prihrávkov pomocou neurónových sietí. No nepodarilo sa im navrhnuť prístup, ktorý by bol efektívnejší ako už používaná neurónová sieť.

2.7.2 Okolie ohrozeného spoluhráča

V zjednodušenej predstave ide o analyzovanie len protihráčov, ktorí mohli ovplyvniť a ohroziť nahrávku danému hráčovi. Neanalyzujú sa tak zbytočne aj protihráči, ktorí nemôžu nášho hráča ohroziť.

2.7.3 Driblovanie

Pôvodné schopnosti driblovania tvorcovia tímu FIITMEDIA vylepšili o tzv. nakopávané driblovanie. Pri driblovaní, ak má hráč pred sebou dostatočne veľkú voľnú plochu, je výhodné si loptu nakopnúť na väčšiu vzdialenosť. Preto bol pridaný mechanizmus, ktorý zmapuje okolie hráča a rozhodne, či je vhodné použiť nakopávané driblovanie.

2.7.4 Komunikácia

Popracovali na komunikácií medzi spoluhráčmi navzájom. Hlavná myšlienka spočíva v oznamovaní hráčom, ktorý ma loptu, svojim spoluhráčom akciu, ktorú sa chystá vykonať.

2.7.5 Záver

Tvorcom tímu FIITMEDIA sa podarilo vylepšiť pôvodného hráča FC Farmári a tým zaslúžene obhájiť prvé miesto na turnaji RoboCup at FIIT'06. Okrem vylepšenej stratégie boli odladené i niektoré chyby a tím nastalo zvýšenie stability. Vďaka tomu, že veľké množstvo času venovali opravám existujúcich chýb, nepodarilo sa im implementovať veľkú časť pôvodne zamýšľaných nových strategických vlastností hráča. V svojej dokumentácii načrtli mnoho dobrých vylepšení, no tie sa už stali len námetom pre ďalšie tímy.

2.8 Loptoši

Tím Loptoši pôsobil na FIIT STU v akademickom roku 2006/2007. Pri svojej práci vychádzal z hráča, ktorého v predchádzajúcom roku implementoval tím FIITMEDIA. Medzi hlavné dôvody, kvôli ktorým sa rozhodli práve pre tohto hráča boli:

- Vynikajúce výsledky v turnajoch.
- Kvalitná dokumentácia.
- Priestor pre ďalšie vylepšovanie.
- Jednalo sa o domáci tím, teda ľahký prístup ku kódu.

Hráč je implementovaný v prostredí Microsoft Visual Studio 6.0, v jazyku C++. Táto voľba bola z veľkej časti podmienená existujúcim vylepšovaným hráčom, ktorý bol rovnako vyvíjaný v tomto prostredí.

Autori sa zaoberali nasledujúcimi oblasťami:

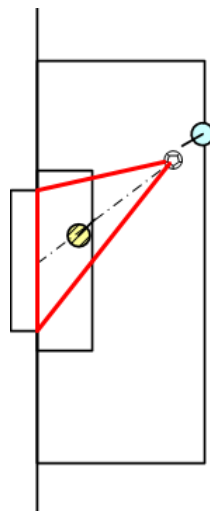
2.8.1 Brankár

V pôvodnom návrhu sa vylepšením samotného brankára nezaoberali. Avšak neskôr zistili, že brankár obsahuje značné množstvo chýb, či už programátorských alebo priamo v jeho rozhodovacej logike. Z tohto dôvodu bolo nevyhnutné, aby sa mu predsa venovali.

Brankár obsahoval viacero chýb, medzi najviditeľnejšie patrilo to, že brankár v určitom počte cyklov nevidel loptu. V pôvodnom riešení sa brankár pri vybehnutí najprv vrátil na svoju domovskú pozíciu ale pritom sa nepozeral okolo seba, či pri ňom nie je lopta. Po návrate potom zväčša dostával gól. Tím Loptoši to vyriešil tak, že brankár sa najprv pozerá okolo seba, či neuvidí loptu a až potom sa vráti na svoju domovskú pozíciu.

Ďalším nedostatkom brankára bolo, že nevedel dostatočne dobre vypočítať stretnutie s loptou, čo sa prejavovalo tým, že loptu nezachytil, keď išla popri ňom alebo sa ju snažil chytiť, keď ešte bola príliš ďaleko. Autorom tímu sa tieto chyby podarilo vyriešiť takým spôsobom, že zaviedli tzv. bezpečnú oblasť, t.j. oblasť vzdialenú od brankára v rozpätí 0 až 1. Brankár sa rozhodne loptu chytiť vtedy, ak je poloha lopty z daného intervalu. Zároveň naprogramovali funkciu, ktorá rozhodovala, či sa lopta môže chytiť o jeden resp. dva cykly skôr.

Poslednou dôležitou zmenou v implementácii brankára bolo zavedenie tzv. správneho držania defenzívnej pozície. Znamená to, že keď si zavedieme trojuholník, ktorého body budú pravá tyčka brány, ľavá tyčka brány a lopta, tak brankár sa musí nachádzať na jeho ťažnici spájajúcej vrchol predstavujúci loptu s jeho protíľahlou stranou. Táto ťažnica prakticky predstavuje úsečku spájajúcu loptu a stred brány. Situácia je znázornená na obr. č. 8.



Obr. 8: Príklad dodržania defenzívnej pozície brankára.

2.8.2 Synchronizácia

Autori tímu Loptoši sa pri skúmaní a štatistickom vyhodnocovaní údajov, ktoré vidia jednotliví hráči, stretli s problémom nedostatočnej synchronizácie údajov v komunikácii so serverom. V hráčovi bola implementovaná synchronizácia za pomoci typu pohľadu (úzky, široký) pomerne komplikovaným a nie správnym spôsobom. Výsledkom tejto implementácie bolo, že hráč veľmi často v nejakom cykle konal skôr ako dostal vizuálnu informáciu.



Pri riešení sa inšpirovali tímom Deravá kopačka, ktorého hráč konal vždy približne 30 milisekúnd pred koncom každého cyklu. Zistili, že server neposiela správu *sense_body* vždy v presne stanovených intervaloch (štandardne 100 milisekúnd), preto museli túto nepresnosť zohľadniť aj pri voľbe ich vlastnej konštanty. Riešenie teda vyzerá tak, že hráč po začiatku cyklu čaká na príchod ďalšej správy. Ak je táto správa *see*, znamená to, že hráč dostal vizuálnu informáciu a môže konať na základe nových údajov od servera. Ak však hráč nedostane správu *see* do stanovenej doby (dĺžka_cyklu_v_milisekundách – 30), rozhodne sa konať na základe starých informácií, pretože je veľká pravdepodobnosť, že vizuálnu informáciu už v danom cykle nedostane.

2.8.3 Vedúci hry

2.8.3.1 Výber kandidáta na vedúceho

Tím riešil otázku výberu kandidátov na vedúceho pomocou nasledovných úvah:

- Vedúci by sa mal obzerať tak, aby mal čo najčerstvejšie informácie o svojom okolí.
- Výber vedúceho sa vykoná na základe kvantitatívneho ohodnotenia jeho informácií o svete.
- Kritériom na výber vedúceho bude vzdialenosť od „centra diania“, ktoré je zvyčajne tam, kde sa nachádza lopta. Musíme tiež zohľadniť šum, ktorý môže vzniknúť pri veľkých vzdialenostiach od vnímaných objektov.

Následne sa rozhodli vykonať vyhodnotenie štatistických údajov o pozícii jednotlivých hráčov, lopty a počte hráčov (súperovho alebo vlastného tímu), ktorých vidí. Tieto dáta ďalej analyzovali pomocou algoritmu na tvorbu rozhodovacích stromov. Následne sa vykonala analýza viacerých možných kombinácií dát a na základe zozbieraných údajov zistili, že najlepší prehľad o hre má hráč, ktorý je od lopty vzdialený minimálne 12 metrov.

2.8.3.2 Zmena vedúceho

Loptoši navrhli a implementovali komunikačný protokol pre zmenu vedúceho hry. Proces zmeny vedúceho hry sa zakladá na nasledujúcom algoritme:

- Priblíženie sa ku kandidátovi.
- Nastaví sa funkcia *AttentionTo*.
- Čaká sa 2 kolá a na základe odpovede sa rozhodnutie akceptuje/neakceptuje.

2.8.3.3 Priradenie právomocí vedúceho vzdialenejšiemu hráčovi

Proces priradzovania právomocí vedúceho vzdialenejšiemu hráčovi sa deje pomocou medzihráča, ktorý sa stane vedúcim ako prvý. Na neho sa spolieha, že najvhodnejšieho vedúceho hry vyhodnotí rovnako správne a rovnakým spôsobom, bez toho, aby sa mu implicitne prikázalo, koho má označiť.



2.8.4 Situácie

Loptoši implementovali rozhranie, pomocou ktorého sa môžu doimplementovať nové situácie. Ako prvú funkciu implementovali možnosť rozpoznať novú správu so situáciou. Podľa typu tejto správy a čísla svojho dresu si hráč nastaví svoju úlohu v tejto situácii.

2.8.4.1 Roh

Pri rohovej situácii Loptoši implementovali dve možnosti zahrana – buď priamo, kopom pred bránu, alebo pomocou rýchlych prihrávkov priviesť loptu pred bránu.

2.8.4.2 Aut

Pri situácii zahrávania autu sa analyzujú tri možné situácie, z ktorej sa vyberie tá, ktorá má najvyššiu pravdepodobnosť spracovania prihrávky. Prihrávka potom nejde po krídle, ale cez stred ihriska a to na základe toho, či sa aut zahráva na súperovej polovici alebo na stredovej čiare. V prvom prípade sa nahrá bližšie k pokutovému územiú súpera, v druhom prípade sa rozohrá podľa zvolenej stratégie (defenzívna/ofenzívna).

2.9 Farmári

FC Farmári je tím z roku 2004/2005, ktorý nadväzoval na prácu tímu z predchádzajúceho roku s názvom Stjupid Dox. Tím podrobne zanalyzoval realizáciu hráča, ktorý bol vytvorený predchádzajúcim tímom a navrhol zlepšenia, ktoré im zabezpečili víťazstvo v súťaži RoboCup at FIIT'05.

2.9.1 Vylepšenia hráča

FC Farmári sa pri návrhu svojho riešenia zamerali najmä na nasledujúce časti vylepšenia hráča.

2.9.1.1 Využitie zrkového vnemu hráča

Systém, ktorý navrhli, udržiava loptu v zornom poli hráča. Pri obzeraní sa hráča dbali na to, aby lopta bola vždy maximálne pri ľavom alebo pravom okraji zorného uhla hráča.

2.9.1.2 Vylepšenie kouča, rozumné umiestnenie hráčov na hracej ploche

Hlavnou úlohou implementovaného kouča je schopnosť meniť hernú taktiku na základe jeho pohľadu na hru ako aj analýza obsadenosti ihriska súperovho tímu. FC Farmári rozdelili hraciu plochu na sektory s veľkosťou 2 x 2 metre. Počas hry sledujú obsadenosť jednotlivých sektorov ihriska a túto informáciu si uchovávajú v dvojrozmernom poli celých čísel. Štatistika sa vyhodnocuje v pravidelných intervaloch. Na základe týchto informácií sa mení rozostavenie:

- Brániacich hráčov na hracom poli tak, aby hráči boli čo najčastejšie v blízkosti sektorov, ktoré sú často obsadzované súperom.
- Útočiaci hráči sa snažia zdržovať v sektoroch, ktoré sú najmenej obsadzované protivráčmi.



2.9.2 Analýza súperových formácií

Na analýzu súperových formácií používajú FC Farmári doprednú neurónovú sieť. Na základe štatistík, ktoré dokážu získať sledovaním obsadenosti sektorov a po zistení formácii súperiaceho tímu, kouč určí vhodnú protiformáciu a vhodnú stratégiu útoku alebo obrany. Predpokladajú, že použitá sieť nebude vykazovať 100 % úspešnosť pri definovaní formácii, avšak každých 5 cyklov vyhodnotia rozmiestnenie hráčov a následne použijú neurónovú sieť, aby formáciu klasifikovala. 10 posledných formácii si zapamätá a prideli im váhy na základe ich aktuálnosti (najmenšiu váhu bude mať najstaršia formácia a naopak).

2.9.3 Výber heterogénnych hráčov

Kouč má možnosť striedania hráčov počas zápasu. Na základe toho ako zanalyzoval hru vie určiť, ktorý hráč z heterogénnych hráčov má najlepšie vlastnosti v danej situácii. V práci je rozpracované aj použitie heterogénnych hráčov a ich presné vlastnosti.

2.9.4 Použitie neurónových sietí na ovládanie nízkych schopností hráča

Tím zvolil Feed Forward sieť s metódou Reinforcement Learning Temporal Difference Alfa, ktorá má schopnosť učiť sa už po vykonaní niekoľkých krokov.

2.9.5 Úprava brankára

Tím pokračoval v práci tímu Stjupit dox aj v brankárovi. Hlavným cieľom vylepšenia brankára je schopnosť odhaliť heterogénneho protihráča a na základe zistenia vhodne reagovať. Brankár nepočíta úspešnosť chytania lopty z konštanty rýchlosti protihráčov, ale zisťuje ich rýchlosť a zo zistených rýchlostí protihráčov určí svoju schopnosť loptu chytiť.

2.9.6 UnitCreator - Tvorba a vyhodnotenie pokrytia subformácií

Tím vytvoril aplikáciu, ktorá sa používa na tvorenie nových jednotiek formácií, resp. subformácií. Používateľ vyznačí pozície hráčov v novej formácii a UnitCreator vygeneruje výstupný súbor. Vo vygenerovanom súbore je uchovaná nová formácia vo formáte, ktorý je možné priamo použiť v zdrojovom kóde.

2.9.7 Záver

Tímu FC Farmári sa podarilo implementovať niekoľko zaujímavých nápadov. Ich zlepšenie brankárovej schopnosti chytať loptu bolo úspešné ako aj systém analyzátora formácií. Ich práca je rozsiahla a všetky navrhované a aj niektoré zrealizované zmeny sú veľmi dobre opísané a vysvetlené. Napriek dobre spracovanej dokumentácii a deklarácii, že implementované časti sú korektné a dobre fungujúce, nasledujúce tímy pokračujúce vo vývoji hráča mali veľké problémy s úpravou nedotiahnutých algoritmov v kóde hráča. Avšak tím sa vo svojom ročníku umiestnil na turnaji RoboCup at FIIT'05 na prvom mieste. Počas súťaže raz prehrál a dvakrát remizoval.



3 Hrubý návrh riešenia

Ako základ pre našu prácu sme si zobrali výsledok tímu Gang Of Six, tým sme od nich prebrali architektúru ich hráča, ktorý vychádza z jadra tímu Uva Trilearn. Dôvodom ich výberu bol homogénny zdrojový kód s kvalitne spracovaným nižšie úrovňovým správaním sa. Reimplementovali vysokoúrovňové riadenie hráča a dobudovali ďalšiu funkcionálnosť, vďaka ktorej sa podarilo spomínanému tímu veľmi dobre uspieť na každoročne organizovanej súťaži RoboCup. Výsledok tejto súťaže sa stal jedným z hlavných kritérií pri výbere tímu, v ktorého práci sme sa rozhodli pokračovať a zlepšovať vlastnosti jeho hráčov. Z predpokladu dokonalosti jadra preberaného hráča sme sa rozhodli nerobiť radikálne zmeny v samotnej architektúre. Hlavným cieľom je oprava drobných nedostatkov a implementácia rozširujúcich vlastností, ktoré by mali zdokonaľiť celkový výsledok.

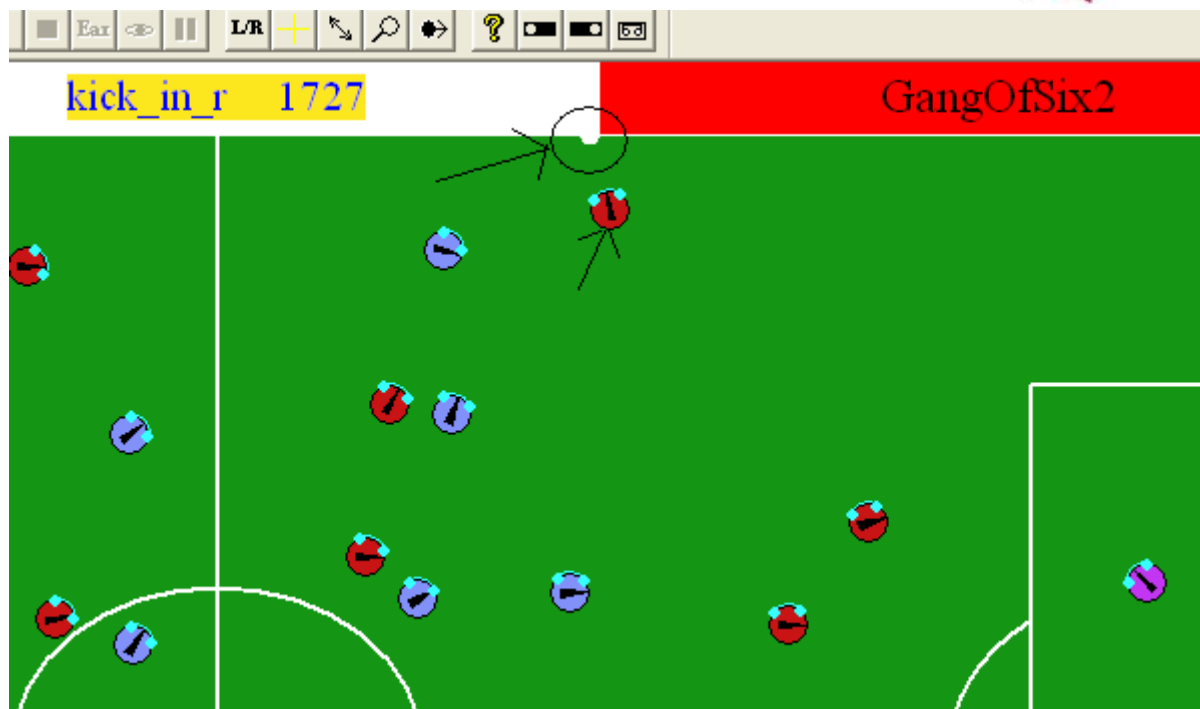
Medzi ciele patrí vyriešenie:

- Problematika zahrávania autových situácií.
- Spôsoby zdokonalenia brankára.
- Zavedenie a implementovanie kouča.
- Vyriešenie možnosti využitia heterogenných hráčov.
- Dopracovanie algoritmu vylučovania premených.

3.1 Zahrávanie autov

Hráč tímu Gang Of Six pomerne často prichádza o loptu z dôvodu nezahratia autu. K strate lopty dôjde kvôli tomu, že hráč, ktorý je najbližšie pri lopte a je poverený zahratím autu, nedokáže loptu nájsť. Hráč nedokáže loptu nájsť aj napriek tomu, že stojí otočený čelom priamo k lopte. Ak tím nevyužije možnosť zahrávať aut v určitom časovom intervale, loptu stratí v prospech súperovho tímu. Táto situácia je obzvlášť závažná, ak tím stratí loptu v blízkosti vlastnej brány. Súper sa takýmto ľahkým spôsobom dostane k výbornej príležitosti streliť gól. Zbytočná strata lopty však zamrzí aj na súperovej polovici ihriska, keďže takýmto spôsobom sa mužstvo pripraví o možnosť zaútočiť na súperovu bránu.

Situácia, keď je hráč otočený čelom smerom k lopte a ani napriek tomu ju nedokáže nájsť je znázornená na obr. č.9. Červený hráč, ktorý je najbližšie pri lopte a poverený zahratím autu, stratí možnosť aut zahratí a k lopte sa dostane modré mužstvo. Obrázok zároveň znázorňuje situáciu, keď sa súperove mužstvo dostane ľahko k lopte na vlastnej polovici ihriska tímu Gang Of Six.



Obr. 9: Strata lopty počas zahrávania autu.

Nájsť zdroj problému a riešenie tejto situácie je pomerne náročné, keďže táto situácia sa neopakuje vždy, ale len v určitých okamihoch. Jediným spôsobom ako nájsť chybu, ktorá spôsobuje tento problém, je detailná analýza zápasov tímu a zdrojového kódu hráča. Počas analýzy zápasov bude potrebné si všimnúť situácie, kedy dochádza k strate lopty týmto spôsobom a pokúsiť sa nájsť okolnosti, ktoré sa vždy opakujú pri tejto situácii. Ak sa podarí nájsť okolnosti, ktoré sa vždy opakujú pri strate lopty, potom bude možné vytipovať časti kódu, ktoré môžu spôsobovať spomenutý problém. Po výbere tých častí kódu, v ktorých sa pravdepodobne nachádza chyba, bude potrebné detailne zanalyzovať zdrojový kód a pomocou rôznych techník ladenia zdrojového kódu nájsť chybu, ktorá zapríčiňuje to, že hráč nedokáže nájsť loptu za autovou čiarou.

3.2 Spôsoby zdokonalenia brankára

Brankár patrí medzi nosné pozície a pre celkový úspech tímu je jeho dokonalosť veľmi dôležitá. Nedostatky spôsobené zlou implementáciou niektorých jeho vlastností môžu a vo väčšine prípadov zapríčinia celkový neúspech. Pri analýze prebratého hráča, v ktorom sme sa rozhodli pokračovať a zdokonaľovať, boli objavené iste nedostatky priamo v tejto pozícii.

Hlavné oblasti zdokonaľovania:

- Chytnie strely smerujúcej popri brankárovi.
- Nový spôsob návratu do bránky pri vybehnutí.
- Vylepšenie rozohrávania.



3.2.1 Chytenie strely smerujúcej popri brankárovi

Myšlienkou tejto časti vylepšovania funkcionality je zdokonalenie vypočítavania stretu brankára s loptou. Týmto sa chce predísť situáciám, že loptu nezachytí, keď ide popri ňom alebo sa rozhodne spraviť príkaz CATCH (chytenie lopty) aj napriek tomu, že lopta bude pred ním ďaleko. Táto chyba môže byť spôsobená nepresným dedukovaním reálnej pozície lopty. Presnejšie povedané, brankár môže vo svojom svete vidieť loptu v svojej blízkosti, no v skutočnosti tam nebude. Preto sa rozhodne vykonať príkaz CATCH, ktorým sa mu nepodarí loptu zachytiť. Riešenie vidíme v zavedení „bezpečného chytania“. Hráč sa rozhodne chytiť loptu vtedy, ak vzdialenosť lopty je parameter servera `catchable_area_l` * k , kde k je konštanta z intervalu $\langle 0,1 \rangle$.

Rozhodli sme sa ďalej implementovať funkcionality, ktorú rozpracovali vo svojom hráčovi členovia tímu Loptoši. Ide o možnosť určenia bezpečného chytania lopty o 1 alebo 2 cykly. Určí sa predpokladaná pozícia lopty o 1 cyklus a simuluje sa, čo sa stane, ak brankár použije príkaz DASH rôznej sily od minimálnej novej až po maximálnu možnú. Takto sa určí, kedy bude hráč najbližšie k lopte v nasledujúcom cykle. Ak bude v nasledujúcom cykle dostatočne blízko lopty, rozhodne sa príkaz DASH odoslať na server.

Týmto istým spôsobom sa určí predpokladaná pozícia o 2 cykly. V tomto prípade sa predpokladá vyskúšanie poslanie v nasledujúcich dvoch cykloch príkaz DASH (ide o kombináciu DASH-DASH) . Ak bude o dva cykly dostatočne blízko lopty, rozhodne sa poslať prvý príkaz DASH na server. V ďalšom cykle bude znova vyhodnocovať, aký príkaz DASH ma použiť. Obdobne sa rozhodne vyskúšať kombináciu príkazov TURN-DASH. (to znamená, že sa otočí príkazom TURN a v ďalšom cykle pošle príkaz na server DASH)

Tento nový spôsob chytania, ktorý bude prebratý z hráča tímu Loptoši a implementovaný v nami zdokonaľovanom hráčovi, zjavne vylepší problémy opísané na začiatku kapitoly.

Predpokladá sa implemetácia vyššie spomínanej funkčnosti a následné testovanie vylepšenia sledovaním celkového prínosu pri samotnom zápase. Niekedy hráč môže vyhodnotiť, že nemusí vykonať žiadnu akciu a lopta k nemu príbehne sama.

Rozhodli sme sa tento prístup použiť pre maximálne dva cykly. Pri väčšom počte cyklov hráč dostáva nepresné súradnice lopty a následné jej chytenie môže byť nebezpečné. Na vypočítanie presnej pozície stretnutia lopty s brankárom sa používa funkcia `GetNearestBallInterceptPos`, ktorá nie je vždy presná.

3.2.2 Nový spôsob návratu do bránky pri vybehnutí

Ďalším veľmi podstatným nedostatkom brankára je nedokonalé vracanie sa na svoju pozíciu v bráne pri vybehnutí z nej. To má za následok inkasovanie veľmi lacného gólu, ktorý môže byť zbytočnou príčinou celkového neúspešného koncového výsledku. Túto slabinu môžu súperovi veľmi ľahko využiť vo svoj prospech. Myšlienkou tejto časti je implementovanie dokonalého sa vracania založeného na aktuálnom určení pozície a držaní si defenzívneho stavu.

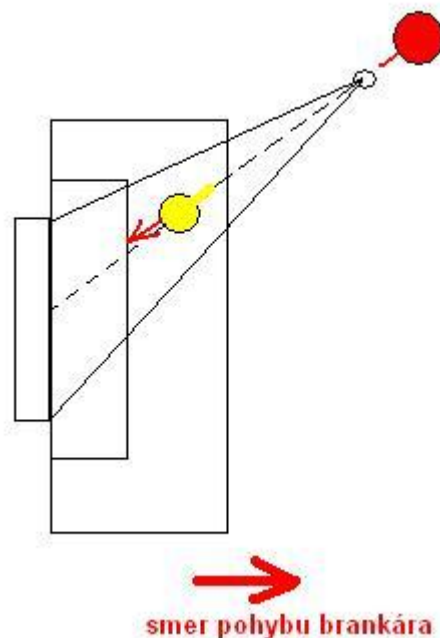
Rozdelenie návratu do bránky na základe aktuálnej pozície:

- Brankár sa nachádza v šestnástke.
- Brankár sa nachádza mimo šestnástky.

3.2.2.1 Brankár sa nachádza v šestnástke

Simuluje sa situácia, že brankár vybehne za loptou, ktorú sa mu nepodarí dobehnúť, pretože sa jej zmocní súperov hráč. Brankár sa za normálnych okolností otočí, čo stojí istý počet taktov a vracia sa otočený chrbtom k hráčom späť na svoju pozíciu v bráne. Týmto spôsobom nevidí, čo sa deje za jeho chrbtom a stráca prehľad o hre (pozri obr. č.10). Je to interval, v ktorom je skoro sto percentná úspešnosť strelenia gólu. Ide o nežiaduci stav, ktorému sa snažíme zabrániť. Preto sme sa rozhodli zmeniť štýl vracania sa späť na aktuálnu pozíciu.

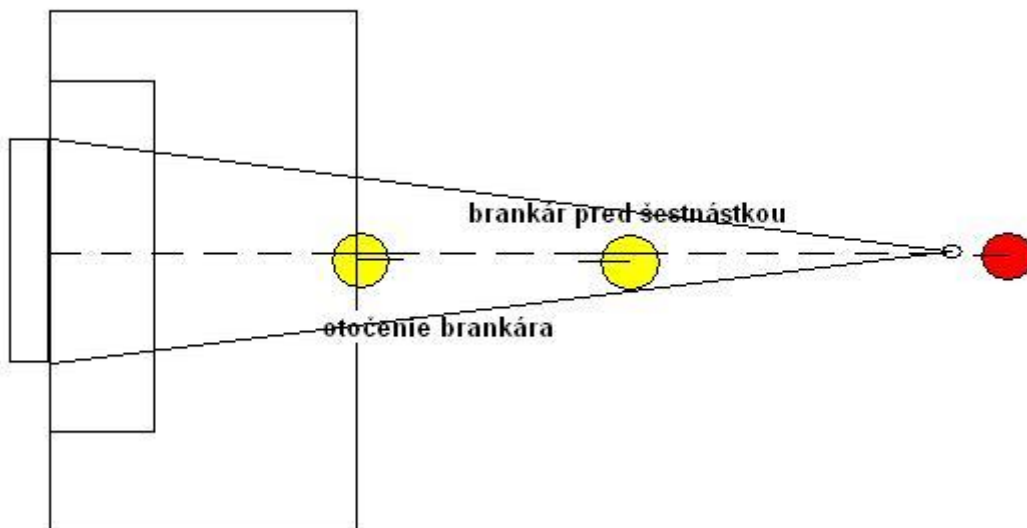
V prvom rade sa zisti, či sa brankár nachádza v šestnástke. Ak sa nachádza v spomínanom území, tak sa neotáča ale začne pomaly cúvať späť na aktuálnu pozíciu a neustále sleduje hru a snaží sa držať defenzívnu pozíciu voči útočiacemu hráčovi. Pojem defenzívna pozícia je vysvetlený v ďalšej kapitole. Týmto spôsobom sa podarí brankárovi ušetriť istý „čas“, ktorý by stratil pri otáčaní. Hlavnú výhodu vidíme v tom, že hráč ostatne stále nasmerovaný v smere hry. Pre hráča to znamená prehľad o aktuálnom stave. Hlavnou nevýhodou implementovanej funkčnosti je, že cúvanie späť je oveľa pomalšie ako keď brankár uteká otočený smerom k danému miestu. Touto funkčnosťou chceme odstrániť vyššie spomínané nežiaduce správanie.



Obr. 10: Obranný pohyb brankára v šestnástke.

3.2.2.2 Brankár sa nachádza mimo šestnástky

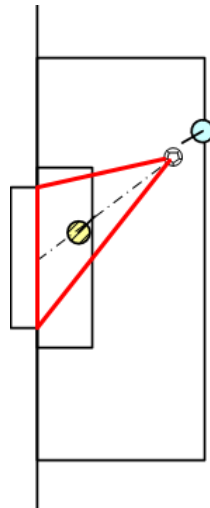
Ak nastane situácia, že sa brankár ocitne mimo územia šestnástky a nedostane sa k lopte, nasleduje situácia vracania sa späť na svoje aktuálne miesto do bránky. Tento proces je rozdelený na dve základné časti. Prvá je spôsob návratu do územia šestnástky, ktoré prebieha jednoduchým spôsobom. Brankár sa rozhodne vrátiť späť. Zistí, že je mimo územia šestnástky z aktuálnej pozície. Nasleduje otočenie a uteká chrbtom otočený k smeru hry k spomínanému územiu. Ak sa dostane na rozhranie šestnástky otočí sa a pokračuje presne podľa správania sa opísaného v predchádzajúcej časti, ktorá pojednávala o obrannom správaní sa hráča, ak sa nachádza v spomínanom území. Predpokladáme, že takáto situácia nenastane. Ide skôr o dorobenie bezpečnostnej funkčnosti, ktorá by ošetrovala aj takýto možný prípad. Na obr. č. 11 je znázornená vyššie opísaná situácia.



Obr. 11: Obranný pohyb brankára mimo šestnástky

3.2.3 Držanie defenzívnej pozície

Je veľmi dôležité, aby sa brankár vždy nachádzal na takej pozícii, že pravdepodobnosť inkasovania gólu strelením lopty k pravej alebo k ľavej tyčke bránky je rovnaká. Toto sa dosiahne správnou pozíciou brankára. Spomínaná situácia je zobrazená na obr. č.12. Pri útoku na bránku jedným súperovým hráčom je potrebné, aby žiadna strana bránky nebola odkrytá. Z toho je jasne, že ak si spravíme trojuholník tým, že sa spoja tyčky bránky s loptou, tak brankár sa musí nachádzať na ťažnici tohto trojuholníka, kde vrchol predstavuje lopta. Inými slovami brankár leží na úsečke, ktorej jeden koniec smeruje do stredu bránky a druhý končí na lopte útočiaceho súpera. Týmto prístupom sa eliminuje pravdepodobnosť odkrytia jednej strany bránky, zníži sa celkové riziko strelenia gólu súperovým hráčom a tým sa vytvára možnosť celkovo lepšieho konečného výsledku.



Obr. 12: Príklad dodržania defenzívnej pozície brankára.

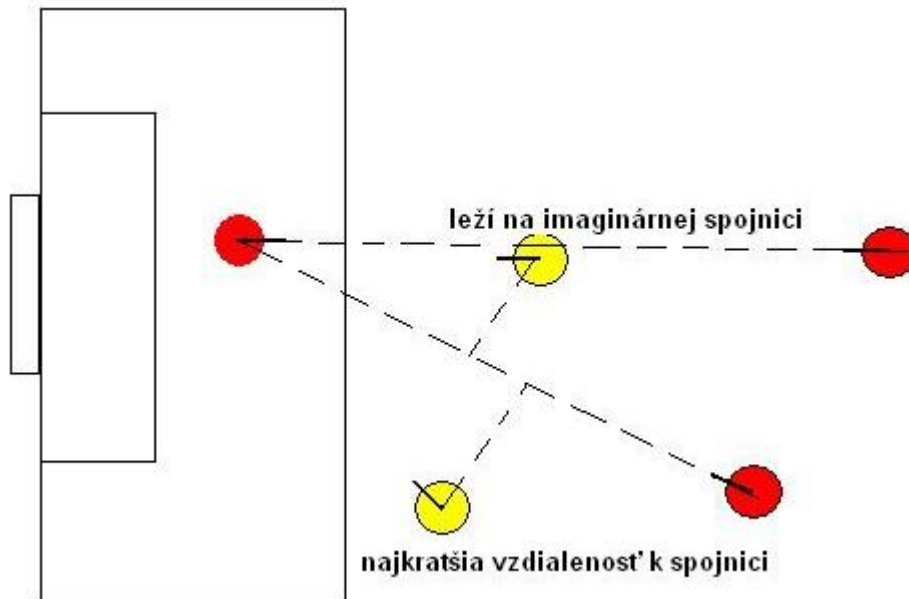
3.2.4 Vylepšenie rozohrávania

Prvá chyba alebo nedostatok v samotnom preberanom návrhu finálneho hráča od tímu Gang of Six bol spôsob rozohrávania lopty samotným brankárom. Pri rozohraní dochádza vo väčšine situácií k zlému adresovaniu prvotnej prihrávky, čo má za následok stratu lopty a zároveň vytvorenie pozície pre súperov tím streliť gól.

Predpokladá sa, že chyba je spôsobená nedokonalým dokončením funkčnosti rozohrávania tejto štandardnej situácie. Došli sme k záveru, že brankár analyzuje situáciu a rozhodne sa nahráť loptu svojmu spoluhráčovi. Vytvorí sa imaginárna spojnica, ktorá predstavuje spojenie brankára a lopty s hráčom, ktorému je prihrávka adresovaná. Potom nasleduje samotná prihrávka, bez ohľadu na to, či sa na danej imaginárnej spojnici nachádza alebo nenachádza protivráč. Týmto spôsobom a nedostatkom v samotnej funkčnosti dochádza k zbytočným stratám lôpt.

Hlavným cieľom je vytvorenie funkcie *controlYourPass*, ktorej hlavným cieľom bude kontrolovať, či sa na danej spojnici alebo v jej blízkosti sa nenachádza protivráč, ktorý by mohol danú prihrávku chytiť. Ak sa protivráč bude nachádzať mimo spojnice, bude sa prerátavať pravdepodobnosť stretu s loptou a možnosťou jej zachytenia (pozri obr. č. 13). Zjednodušene povedané, vyráta sa, či sa daný protivráč má možnosť v danej blízkosti dostať na úroveň imaginárnej spojnice skôr, než lopta k adresátovi. Vzdialenosť medzi protivráčom a imaginárnou spojnicou predstavuje imaginárna kolmica na spojnicu brankára a jeho spoluhráča. Ak funkcia vráti výsledok, že protivráč je schopný dostať sa na spomínanú pozíciu, tak sa táto možnosť berie ako neprípustná. Berie sa do úvahy ďalší spoluhráč, pri ktorom sa postupuje presne tým istým spôsobom. Týmto sa vytvorí fiktívny rebríček pravdepodobností možnosti rozohrania štandardnej situácie. V prípade, že žiadna možnosť nie je vyhovujúca, berie sa prvá z rebríčka možností s najvyššou úspešnosťou prihrávky.

Ak sa na spojnici nachádza protivráč, tak táto možnosť je automaticky zamietnutá a zároveň sa nedostáva ani do rebríčka potenciálnych možností.



Obr. 13: Analýza prihrávky.

3.3 Zavedenie a implementácia kouča

3.3.1 Motivácia

Potreba analýzy hry hráčom si vyžaduje, aby hráč vedel predpovedať, čo sa udeje v jeho okolí v niekoľkých ďalších cykloch. Zvyčajne hráč vie predpovedať ako rýchlo k nemu príbehne protihráč alebo kde sa môže protihráč nachádzať. Problém ale nastane, ak hráči budú mať niektoré vlastnosti, ktoré ich odlišujú od ostatných hráčov. Takýmito vlastnosťami sú napríklad rýchlosť, zrýchlenie, rotácia hráčov alebo dosah lopty. Ak niektorí súperovi hráči disponujú týmito vlastnosťami, je pre nášho hráča veľmi ťažké predpokladať ako sa zachová súper v ďalších cykloch. To môže mať za následok častú stratu lopty a tým aj väčšiu šancu pre súpera že strelí gól.

Zisťovanie súperových heterotypov našim hráčom prináša niekoľko problémov. Medzi najpodstatnejšie podľa nás sú:

Hráč má nepresné informácie spôsobené šumom, ktorý do hry pridáva server.

- Aby hráč mohol zistiť vlastnosti súperových hráčov, potrebuje sa na nich neustále pozerať, čo by mu neumožnilo sa veľmi zapájať do hry.
- Svet hráča by sa stal príliš zložitý ak by malo do toho prísť aj uvažovanie a zisťovanie súperových heterotypov.

Z týchto dôvodov sa javí ako najrozumnejšie riešenie na zisťovanie súperových heterotypov delegovať túto funkciu na kouča.



3.3.2 Úloha kouča

Kouč môže pomáhať hráčom pri hre. Zvyčajne analyzuje hru, môže rozhodnúť o zmene taktiky zápasu ale taktiež môže podávať hráčom najrôznejšie informácie. Kouč má na rozdiel od hráčov pohľad na ihrisko bez šumu a vidí súčasne celé ihrisko. K analýze hry, a teda aj taktiky, môže koučovi slúžiť aj prístup k logovacím súborom z iných hier. My od kouča budeme vyžadovať plnenie nasledujúcich úloh:

- Zisťovanie súperových heterotypov.
- Odoslanie informácií o týchto heterotypoch vlastným hráčom.
- Nasadenie vlastných heterotypov do hry.

3.3.3 Rozpoznávanie heterotypov

Predpokladom toho, aby mohol kouč zistiť ktorý hráč je heterotyp, je prejavenie sa konkrétnych vlastností tohto hráča. Je predpoklad, že ak hráč má tieto vlastnosti, tak sa niekedy prejavia. Základnou úlohou kouča je sledovanie súperových hráčov a ich jednotlivé charakteristiky ako je rýchlosť, zrýchlenie, rotácia a dosah na loptu.

Server pred začiatkom hry náhodne vygeneruje heterotypy a odošle informácie o nich koučovi. Keďže kouč pozná fyzikálny model správania súperových hráčov a vie zistiť kedy ich správanie v niektorých ohľadoch prekračuje fyzické obmedzenia bežného hráča, môže pomerne jednoducho odhaliť a správne zaradiť daného hráča ako určitý heterotyp.

Informácie prichádzajú koučovi ako vizuálna informácia zo servera. Táto informácia obsahuje polohy jednotlivých hráčov a lopty spolu s ich rýchlosťami na konci simulačného cyklu [1]. Z týchto informácií a informácií, ktoré prišli v predchádzajúcich cykloch, je možné vypočítať charakteristiky jednotlivých hráčov. Rýchlosť hráča sa dá zistiť podľa zmeny polohy oproti predchádzajúcemu cyklu. Rýchlosť na konci každého cyklu je znížená o hodnotu *player_decay*, ktorá je pre každý heterotyp iná. To spôsobuje nejasnosť výsledkov, ak berieme rýchlosť iba ako veličinu priamo závislú na rozdiel umiestnenia hráča v dvoch cykloch. Podobným spôsobom je možné vyrátať aj rýchlosť otáčania.

Zisťovanie vzdialenosti hráča, od ktorej je schopný kopnúť do lopty je pomerne náročná úloha. Keďže kouč nedostane explicitnú informáciu, že do lopty bolo kopnuté, dá sa to zistiť iba podľa toho, že sa zmenila rýchlosť a smer lopty. Tento spôsob, založený na zmene vektoru lopty, je možné použiť iba v prípade, ak sa pri lopte nachádza iba jeden hráč. Ak ich je viac, bolo by problémom zistiť, ktorý hráč kopol do lopty.

3.3.4 Implementácia kouča

Nakoľko je implementácia kouča pomerne rozsiahla úloha, je predpoklad že použijeme už vytvoreného hráča, pravdepodobne vytvoreného tímom Sklo. Tento kouč rozoznáva heterotypy na základe rýchlosti a zrýchlenia hráčov. Pri rozpoznávaní vlastností ako rýchlosť rotácie a dosah na loptu však tento kouč narazil na nemožnosť určiť tieto vlastnosti. Dôvodom bol prílišný šum, ktorý do hry vnáša server. Podarilo sa však empiricky zistiť hraničné



hodnoty a rýchlosti a zrýchlenia, pri prekročení ktorých sú detekované výrazne rýchle heterotypy.

Kouč vytvorený tímom Sklo dokáže zistiť výrazne rýchle heterotypy a po ich rozpoznaní odošle hráčom informáciu o rozpoznaných heterotypoch. Pred nami stojí úloha správne implementovať hráča vytvoreného tímom Sklo a upraviť vnútorný svet našich hráčov, aby boli schopný správne použiť získane poznatky od trénera. Týmto krokom by sa mal naplniť jeden z hlavných cieľov motivácie, presnejšie predpokladanie správania protihráčov.

V tomto koučovi ostávajú viaceré možnosti ako vylepšiť jeho funkcionality. Je vhodné sa popasovať a pomerne presne uhádnuť hodnotu *player_decay*, čo by nám umožnilo presnejšie určiť rýchlosť hráča a tým aj skôr rozpoznať dané heterotypy. Tým Sklo začal pracovať na funkcii, ktorá sa snaží určiť túto hodnotu, je potrebné ju dokončiť a odladiť. Taktiež je potrebné pokračovať a snažiť sa určiť pri hráčovi rýchlosť rotácie a vzdialenosť z ktorej dokáže kopnúť do lopty. Hoci tým Sklo v tejto úlohe neuspel, je vytvorená funkcionality ktorá sa tomu venuje.

Používanie vlastných heterotypov má na starosti kouč. Nakoľko táto funkcionality nie je v nami zamýšľanom koučovi implementovaná, je potrebné ju doplniť. Podľa nami nastavenej konfigurácie kouč na začiatku zápasu postaví heterotypy. V priebehu hry už kouč vymieňať hráčov nebude.

3.4 Heterogénni hráči

Na začiatku futbalového zápasu má kouč možnosť vybrať si zo siedmych typov hráčov, ktorí majú rôzne vlastnosti. Tieto typy hráčov značia, ktoré vlastnosti hráča sú vylepšené a naopak, ktoré zase degradované. Každý heterogénny hráč má niektorú schopnosť na vyššej úrovni (napr. výdrž, zrýchlenie, presnosť nahrávky, sila kopu a podobne) a zároveň niektorú potlačenú. Štandardným typom hráča, ktorý má všetky vlastnosti na jednej úrovni, je hráč typu 0. Tento druh hráča je využívaný tímom Gang of Six, a my pokračujeme v jeho vylepšovaní. Ďalšími možnosťami výberu sú typy od jedna až po typ šesť vrátane. Typy s vylepšenými vlastnosťami sa používajú pri špecifických rolách v tíme. Družstvo využívajúce heterogénny typ hráčov sa prejavuje väčšou rôznorodosťou hry a vo väčšine prípadov je takéto mužstvo úspešnejšie ako tím zložený len z homogénnych hráčov. Kľúčom k úspechu je správny výber typu pre rolu, ktorú hráč bude na ihrisku zastávať.

3.4.1 Vlastnosti hráčov v rôznych roliach

Každý futbalový hráč reálneho sveta je v nejakej roly, ktorá je reprezentovaná konkrétnymi vlastnosťami. Na základe skúseností sme vyšpecifikovali nasledujúce požadované vlastnosti a tie sme prideliť k roly hráča. Údaje sme uviedli do prehľadnej tabuľky (viď tab. č.2: Pridelenie vlastnosti hráčskym roliam).

Tab. 2: Pridelenie vlastnosti hráčskym roliam.

Rola hráča	Požadované vlastnosti
Obranca	Má za úlohu zastaviť prenikajúceho hráča, a preto by mal vedieť vyvinúť veľkú rýchlosť, a musí loptu odkopnúť preč od vlastnej brány, preto musí vedieť kopnúť loptu veľkou silou, avšak nepotrebuje vedieť loptu odkopnúť presne.
Stredopoliar	Prijíma loptu z obrany a rozohráva ju útočníkom, preto musí vedieť presne prihrávať a mať veľmi kvalitné informácie o aktuálnom svete. Nemusí byť rýchly (mať veľkú akceleráciu) a ani obratný. Mal by však mať veľkú výdrž, keďže cez stredopoliaru sa prelína celá hra, od obrany k útoku.
Stredný útočník	Prijíma loptu od stredopoliarov, musí sa s ňou dostať ku bránke a kopnúť gól, preto by mal vedieť vyvinúť väčšiu rýchlosť (akcelerácia) a mal mať veľkú silu kopu. Takisto musí byť obratný, aby ho obrancovia ľahko neobrali o loptu. Tieto vlastnosti môžu byť zvýraznené na úkor presnosti prihrávky a takisto výdrže.
Útočiaci krídelník	Prijíma loptu od stredopoliarov a prihráva ju ďalšiemu útočníkovi, preto musí mať veľkú akceleráciu a spomalenie (na presné zachytenie prihrávok).

Úvahy o potrebných vylepšeniach hráča, ktorý zastáva určitú rolu v tíme, sme zadefinovali do tab. č. 3: Vyžadované vlastnosti pre rolu hráča.

Tab. 3: Vyžadované vlastnosti pre rolu hráča.

Vlastnosť	Vyžadovaný stupeň			
	++++	+++	++	+
Sila kopu	stredný útočník	obranca	útočiaci krídelník	stredopoliar
Rýchlosť hráča	obranca	stredný útočník	útočiaci krídelník	stredopoliar
Akcelerácia	obranca	útočiaci krídelník	stredný útočník	stredopoliar
Výdrž	stredný útočník	útočiaci krídelník	obranca	stredopoliar
Obratnosť	obranca	stredný útočník	útočiaci krídelník	stredopoliar
Presnosť kopu	útočiaci krídelník	stredopoliar	stredný útočník	obranca



3.4.2 Prečo použiť heterogénnych hráčov?

Máme k dispozícii silnú zbraň rôznorodosti hráčov. Ako už bolo načrtnuté vyššie, hráč tímu *Gang of Six* sa nezaoberal využívaním heterogénnych typov hráčov. Preto našou snahou je zlepšenie aj tejto oblasti hráča. Zo záverečnej práce Valla [1] vyplýva, že použitie hráčov výrazne ovplyvní hru tímu v počte kopov na bránu, čase strávenom na súperovej strane ihriska, držaní lopty a aj počte úspešných prienikov do priestoru súpera a k súperovej bráne. Jeho tvrdenie potvrdzuje aj výsledky holandského tímu *UvA Trilearn 2002*, ktoré poukazujú na vyššiu úspešnosť tímu s použitím heterogénnych hráčov ako tímu bez použitia heterotypov.

Využívanie heterogénnych hráčov je možné využitím futbalového serveru verzie 7.x a vyššie. Futbalový server na začiatku hry vytvorí nastavený počet náhodných typov hráčov. Jednotlivé typy hráčov majú rozličné vlastnosti, založené na kompromisoch definovaných v konfiguračnom súbore *player.conf*. Obidva tímy používajú pri vzájomnom zápase tie isté typy hráčov. Typ 0 je stále rovnaký.

Hráči dostanú po pripojení k serveru informáciu o dostupných typoch. Kouč pred začatím hry podľa určitých kritérií, ktoré si zadefinujeme, prideli hráčom ich heterotypy. Počas hry môže kouč zmeniť typy hráčov použitím príkazu *change_player_type subs_max* krát.

3.4.3 Využitie heterogénnych hráčov

Tím FC Farmári sa snažil využiť výhody ponúkané heterogénnymi hráčmi. Pred začiatkom zápasu kouč rozdelil jednotlivým hráčom vlastnosti na základe úvahy, ktorá bola postavená na rozmiestnení hráčov na hracej ploche a na základe úloh hráča v tíme. Našou snahou bude zistiť, či algoritmus pridelenia heterogénneho typu je efektívny a korektný, resp. či jeho použitie zlepšuje konečný výsledok zápasu. Taktiež sa zameriame na správne rozmiestnenie hráčov v zostave podľa ich vylepšených vlastností, ale uvažovať samozrejme budeme aj so slabšou stránkou hráčov. Najvhodnejším spôsobom na využitie rôznorodých hráčov v hre nášho tímu je maximálne možné priradenie potrebných vlastností k ponúkaným možnostiam hráčov, a tak urobiť hru efektívnejšou a viac pripodobnenú k reálnemu svetu.

Po doplnení hráča o schopnosť používať heterogénne typy budeme simulovať zápas medzi pôvodnou verziou hráča a vylepšenou verziou. Uvidíme, či pridanie a použitá logika výberu heterogénnych hráčov je efektívna alebo nie je, predpokladáme však pozitívny výsledok.

3.5 Koordinačné grafy a algoritmus eliminácie premenných

3.5.1 Úvod

Metóda rozhodovania pomocou koordinačných grafov sa používa na vysoko úrovňové rozhodovanie v kooperatívnych multiagentových systémoch, v ktorých všetci agenti majú jeden spoločný cieľ. Koordináciou sa rozumie proces, ktorý zabezpečuje, že individuálne rozhodnutia agentov vyústia v danom okamihu do realizácie optimálnej spoločnej akcie všetkých agentov.



V koordinačnom grafe, každý uzol reprezentuje agenta, každá hrana indikuje, že príslušní agenti spojení hranou musia navzájom koordinovať svoje akcie, ktoré sa chystajú vykonať. Na riešenie lokálnych koordinačných problémov medzi agentami sa používa *algoritmus eliminácie premenných*. Tento algoritmus iteratívne rieši lokálne koordinačné problémy, výsledky riešenia týchto problémov šíri ďalej po koordinačnom grafe. Tieto čiastkové výsledky ovplyvňujú voľbu akcií a riešenie súvisiacich koordinačných problémov medzi ďalšími agentami.

Topológia grafu sa v každom okamihu aktualizuje podľa aktuálnych informácií získaných z modelu sveta hráča. Následne je použitý algoritmus eliminácie premenných na vyriešenie koordinačného problému všetkých agentov.

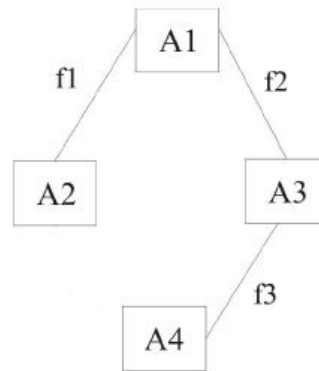
Uva Trilearn použitý algoritmus rozhodovania pomocou koordinačných grafov ešte zjednodušuje tak, že prideli jednotlivým agentom role a namiesto koordinovania akcií jednotlivých agentov koordinuje akcie pre jednotlivé role. Pridelenie rolí agentom redukuje množinu dostupných akcií pre agenta a tak znižuje veľkosť priestoru spoločnej akcie všetkých agentov. Navyše, pri tomto procese, použitím modelu sveta každého agenta a istých spoločných vedomostí dostupných každému agentovi, odpadá nutnosť priamej komunikácie agentov.

Ako bolo spomenuté, tím Gang of Six vychádzal z hráča Uva Trilearn 2003. Keďže však tím Uva Trilearn nezverejnil kompletne zdrojové kódy svojho hráča, musel si tím Gang of Six naimplementovať algoritmus eliminácie premenných sám.

Veľmi podrobný opis problematiky je popísaný v dokumentácii tímov Gang of Six a Uva Trilearn 2003, preto sa v nasledujúcich statiach budeme venovať len stručnému popisu potrebnému na pochopenie problému.

3.5.2 Koordinačné grafy

Pomocou koordinačných grafov vieme koordinovať akcie jednotlivých agentov. V koordinačných grafoch sú uzly jednotliví agenti, ktorí koordinujú svoje akcie. Hrany špecifikujú tzv. výnosové funkcie, ktoré vyjadrujú koordinačné závislosti medzi agentmi. Koordinačná závislosť medzi dvoma agentmi znamená, že to, akú akciu vykoná jeden agent v danom cykle závisí od zamýšľanej akcie, ktorú plánuje vykonať druhý agent (obr. č.14). Lokálne výnosové funkcie špecifikujú aký výnos nastane, ak sa naraz splnia podmienky ňou špecifikované (napr. aký výnos nastane, ak agent A1 vykoná akciu dribluj a zároveň agent A2 vykoná akciu presuň_sa). Príspevky lokálnych výnosových funkcií sa spočítajú do výnosu celkovej spoločnej akcie agentov v danom cykle – globálneho výnosu špecifikovaného globálnou výnosovou funkciou. Vzhľadom na zložitosť hľadania optimálnej spoločnej akcie sa nekoordinujú akcie všetkých agentov navzájom, ale iba akcie podmnožín jednotlivých agentov (pridelením rolí – koordinujú sa role, nie jednotliví agenti).



Obr. 14: Koordinačný graf pre problém koordinácie akcií štyroch agentov.

Globálna výnosová funkcia $R(a)$ môže byť dekomponovaná na lineárnu kombináciu lokálnych výnosových funkcií, z ktorých každá vyhodnocuje spoločnú akciu menšieho počtu agentov ako pôvodná funkcia. Napríklad, predpokladajme, že máme štyroch agentov a ich globálnu výnosovú funkciu môžeme dekomponovať nasledovne:

$$R(a) = f_1(a_1, a_2) + f_2(a_1, a_3) + f_3(a_3, a_4)$$

Jednotlivé funkcie f_i sú tiež výnosové funkcie, ale vyhodnocujú spoločnú akciu iba dvoch agentov a zároveň každá špecifikuje len lokálne koordinačné závislosti medzi menším počtom agentov ako globálna výnosová funkcia. Zodpovedajúci koordinačný graf je zobrazený na obrázku č. 1. Uzol v tomto grafe reprezentuje agenta A_i , lokálne výnosové funkcie f_i špecifikujú lokálne koordinačné závislosti – agent A1 musí koordinovať svoju akciu s agentom A2 a A3. Agent A4 s A3. A3 musí koordinovať svoje akcie s A1 a A4 a nakoniec agent A2 s A1. Globálny koordinačný problém je teda nahradený istým počtom lokálnych koordinačných problémov, ktoré sa týkajú menšieho množstva agentov a ktoré je preto možné jednoduchšie vyriešiť.

3.5.3 Algoritmus eliminácie premenných

Na vyriešenie lokálnych koordinačných problémov sa používa algoritmus eliminácie premenných. Hlavnou myšlienkou je, že agenti sú eliminovaní jeden za druhým po vykonaní tzv. lokálneho maximalizačného kroku, pri ktorom uvažujeme všetky možné kombinácie akcií agentových susedov v koordinačnom grafe.

Algoritmus pracuje nasledovne:

1. Vyberieme náhodne agenta z koordinačného grafu na elimináciu. Tento agent zozbiera výnosové funkcie od svojich susedov.
2. Agent optimalizuje svoje rozhodnutie akú akciu vykoná v nasledujúcom kroku tak, že uváži všetky možné kombinácie akcií svojich susedov, tieto kombinácie akcií spolu so svojou akciou ohodnocuje výnosovou funkciou. Vyberie si takú akciu, ktorá maximalizuje výnosovú funkciu. Následne výsledok – správu o tom, akú akciu vykoná vo forme konštrukcie novej výnosovej funkcie odošle späť svojim susedom.



3. Agent je eliminovaný z grafu.
4. Proces pokračuje bodom 1 dotedy, pokiaľ nie sú z grafu eliminovaní všetci agenti. Ak už sú všetci agenti eliminovaní, pokračuje sa bodom č. 5.
5. Posledný agent vyberie najoptimálnejšiu akciu, ktorá maximalizuje výnosovú funkciu pre celú skupinu agentov v grafe.
6. Vykoná sa prechod grafom v reverznom poradí eliminácie agentov z grafu, aby každý agent vedel určiť optimálnu akciu, podľa zvolených akcií svojich susedov.

3.5.4 Role agentov

Ako bolo spomenuté v časti 2.6 *Analýza hráča tímu Gang of Six*, tím Gang of Six nevytváral graf z agentov, ale z ich rolí. K tomuto rozhodnutiu dospel už tím Uva Trilearn, aby zvýšil výkon algoritmu. Rola sa dá chápať ako množina akcií, ktorú môže daný hráč vykonať. Jednotlivým hráčom sa prideliť rola podľa ich vzdialenosti od lopty. Tím sa zníži množstvo akcií, ktoré by ináč vstupovali do algoritmu eliminácie premenných.

Tím Uva Trilearn 2003 rozlišuje tieto role:

- Active – očakáva sa, že hráč vykoná akciu s loptou
 - passer – hráč môže kopnúť do lopty
 - interceptor – hráč nemôže kopnúť do lopty
- Receiver – očakáva sa, že hráč dostane prihrávku
- Passive – hráč sa presúva na svoju strategickú pozíciu
- Brankár – defaultne priradená rola.

3.5.5 Stav implementácie rozhodovania pomocou koordinačných grafov

Problém hľadania optimálnej spoločnej akcie pomocou metódy koordinačných grafov pozostáva z viacerých častí úloh:

- ✓ Realizovať pridelenie rolí hráčom,
- ✓ Navrhnuť vhodné dátové štruktúry na realizáciu výnosových funkcií,
- ✓ Realizovať tímovú stratégiu pomocou výnosových pravidiel,
- ✓ Implementovať viaceré chýbajúce predikáty vyskytujúce sa vo výnosových funkciách,
- ✓ Implementovať algoritmus eliminácie premenných.

Počas doby riešenia projektu sa tímu Gang of Six úspešne podarilo implementovať algoritmus pridelenia rolí pomocou upravenej Newtonovej metódy. Navyše vytvorili framework pre riešenie problému hľadania optimálnej spoločnej akcie prostredníctvom metódy koordinačných grafov. Stratégiu tímu špecifikovali prostredníctvom deviatich všeobecných



výnosových funkcií (pozri kap. č. 3.6 dokumentácie tímu Gang of Six) použitý v tíme Uva Trilearn 2003.

Hľadanie optimálnej spoločnej akcie spĺňajúcej Nashovu rovnováhu s ohľadom na heuristiku špecifikovanú výnosovými funkciami realizujú pomocou algoritmu eliminácie premenných. Počas letného semestra sa im podarilo implementovať prvú časť algoritmu (dopredný prechod – vyriešenie všetkých lokálnych koordinačných závislostí), druhá časť (spätný prechod – nájdenie globálnej optimálnej spoločnej akcie) nebola dokončená. Celkovo tak úspešne implementovali pridelovanie rolí pomocou rozšírenej Newtonovej metóde, našli spôsob ako reprezentovať výnosové pravidlá, definovali kompletnú tímovú stratégiu a logické rozhodovanie, doplnili množstvo chýbajúcich predikátov, našli vhodný spôsob ako reprezentovať všeobecné akcie. Všetky funkcie sú dostupné v implementácií. Aby bolo možné uvedený prístup vysokoúrovňového rozhodovania nasadiť, je nutné ešte dokončiť druhú časť algoritmu eliminácie premenných.

Teda cieľom nášho tímu pre tento bod je dokončenie algoritmu eliminácie premenných. Algoritmus je dobre zdokumentovaný v dokumentácii tímu Gang of Six, čo bol aj náš hlavný zdroj informácií o tejto problematike.

Naším cieľom v najbližšom období bude v prvom rade otestovať naimplementované časti algoritmu. Tím Gang of Six totiž v dokumentácii nikde neuvádza ako, a či vôbec hotové časti testovali. Po overení funkcií pristúpime k doimplementovaniu algoritmu podľa špecifikácie v ich dokumentácii.



4 Záver

V predkladanom dokumente sme uviedli analýzu mnohých tímov. Tím sme nadobudli pomerne veľké množstvo poznatkov a spôsobov implementácií hráčov. Na každom z analyzovaných hráčov tím našiel pozitíva aj negatíva. Po zvážení všetkých týchto informácií sa tím rozhodol pokračovať vo vývoji hráča tímu Gang Of Six. Hlavnými argumentami pre výber tohto tímu je skutočnosť, že ich hráč zaznamenal na záverečnom turnaji vynikajúce výsledky, v dokumentácii tohto tímu sú všetky aspekty ich hráča podrobne zdokumentovane a zdrojový kód hráča je prehľadný a dobre zdokumentovaný.

V druhej časti dokumentu tím predložil návrhy na vylepšenie funkcionality hráča tímu Gang Of Six. Tím sa rozhodol vylepšiť tie vlastnosti hráča, ktoré sa ukázali po analýze jeho zápasov ako najväčšie problémy. Rozhodli implementovať aj úplne novú funkcionality hráča, ako napr. zavedenie kouča a heterogénnych hráčov. Od navrhnutých a opísaných zmien a vylepšení si tím sľubuje zlepšenie hernej stránky a udržanie vynikajúcich výsledkov aj na tohtoročnom záverečnom turnaji.



Prílohy

V tejto časti sú uvedené prílohy týkajúce sa riadenia projektu. Tu uvádzame ich zoznam:

Príloha A: Bibliografia

Príloha B: Zoznam tabuliek

Príloha C: Zoznam obrázkov



Príloha A: Bibliografia

- [1] Tím FIITMedia. 2006. Analýza, hrubý návrh a prototyp.
<http://www2.dcs.elf.stuba.sk/TeamProject/2005/team02/Dokumentacia2.pdf>. 3.11. 2007.
- [2] Tím Gang Of Six. 2007. Analýza, návrh a špecifikácia.
http://www2.dcs.elf.stuba.sk/TeamProject/2006/team03/public_html/dokumentacia/dokument-final.pdf. 3.11. 2007.
- [3] Tím Gang Of Six. 2007. Implementácia navrhnutého riešenia.
http://www2.dcs.elf.stuba.sk/TeamProject/2006/team03/public_html/dokumentacia/implementacia.pdf. 3.11. 2007.
- [4] ORCA. 2001. Homepage of the AI-studentproject at the University of Osnabrück Wintersemester. <http://www.ikw.uni-osnabrueck.de/~orca>. 3.11. 2007.
- [5] Tím Sklo. 2004. Dokumentácia k softvéru.
<http://www2.dcs.elf.stuba.sk/TeamProject/2003/team08/>. 3.11. 2007.
- [6] Tím Squirrel Squadron. 2006. Dokumentácia k systému.
http://www2.dcs.elf.stuba.sk/TeamProject/2004/team01/download/dokumentacia_system.pdf. 3.11. 2007.
- [7] Vallo, M. 2006. Využitie vlastností heterogénnych hráčov. Záverečný projekt. Bratislava: FIIT STU. 3.11. 2007.



Príloha B: Zoznam tabuliek

Tab. 1: História zmien dokumentu.....	2
Tab. 2: Pridelenie vlastnosti hráčskym roliam.....	41
Tab. 3: Vyžadované vlastnosti pre rolu hráča.....	41



Príloha C: Zoznam obrázkov

Obr. 1: Architektúra hráča tímu UvaTrilearn.	17
Obr. 2: Príklad koordinačného grafu.	21
Obr. 3: Príklad otáčania hlavou	23
Obr. 4: Príklad osobných kruhov hráčov.	24
Obr. 5: Príklad rozhodovania sa pri driblovaní.	25
Obr. 6: Povolená zóna brankára.	25
Obr. 7: Hráč v spodnej časti prihráva spoluhráčovi do behu, pričom prihrávku ohrozuje protihráča 2.	26
Obr. 8: Príklad dodržania defenzívnej pozície brankára.	28
Obr. 9: Strata lopty počas zahrávania auta.	33
Obr. 10: Obranný pohyb brankára v šestnástke.	35
Obr. 11: Obranný pohyb brankára mimo šestnástky.	36
Obr. 12: Príklad dodržania defenzívnej pozície brankára.	37
Obr. 13: Analýza nahrávky.	38
Obr. 14: Koordinačný graf pre problém koordinácie akcií štyroch agentov.	44