

Slovenská technická univerzita v Bratislave

FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLOGIÍ

Ilkovičova 3, 812 19 Bratislava

Tvorba softvérového systému v tíme

RoboCup – nové stratégie

(Dokumentácia k projektu)

Tím č. 2 – FIITMEDIA Bratislava

Bc. Lubomír Hromádka, Bc. Peter Ledňa, Bc. Tomáš Matúšek, Bc. Tomáš Minčeff

Vedúci tímu: Ing. Ivan Kapustík

2005/2006

OBSAH

Zoznam tabuliek.....	viii
Zoznam obrázkov.....	viii
1 Úvod	1
1.1 O dokumente	1
1.2 Zadanie	2
2 Analýza.....	3
2.1 SoccerServer	3
2.1.1 Základné pojmy	3
2.1.2 Analýza servera.....	3
2.1.3 Prehľad verzií SocckerServer	6
2.2 FC Farmári.....	7
2.2.1 Základné informácie.....	7
2.2.2 Analyzátor ihriska v koučovi.....	8
2.2.3 Tvorba a vyhodnotenie pokrytia subformácií	9
2.2.4 Neurónová sieť učiaci sa driblovať s loptou.....	10
2.2.5 Výber spoluhráča na prihrávku na báze učenia s odmenou	10
2.2.6 Úprava brankára tímu Stjupit dox	11
2.2.7 Zhodnotenie.....	11
2.3 Squirrel Squadron.....	11
2.3.1 Základné informácie	11
2.3.2 Zvuková komunikácia	12
2.3.3 Dynamické zmeny formácií	13
2.3.4 Správanie sa hráča	13
2.4 L. A. S. T. United	14
2.4.1 Základné údaje.....	14
2.4.2 Komunikácia	15
2.4.3 Rozoznávajúce situácie na ihrisku	15
2.4.4 Strategické rozmiestňovanie.....	16
2.4.5 Manažment energie hráča.....	16
2.4.6 Skenovanie situácie	16
2.5 Sklo	16
2.5.1 Vnútorný svet hráča	17
2.5.2 Zvuková komunikácia	17
2.5.3 Všeobecné taktické funkcie	17
2.5.4 Plánovaný výkop	18
2.5.5 Taktika hráčov	18

2.5.6	Rozlišovanie heterotypov koučom.....	19
2.5.7	Zhrnutie	19
2.6	Brainstormers	20
2.6.1	Základné informácie	20
2.6.2	Architektúra hráča.....	20
2.6.3	Individuálne stratégie hráčov	20
2.6.4	Určovanie pozície hráčov	21
2.7	UvA Trilearn	21
2.7.1	Základné informácie	21
2.7.2	Analýza hráča	22
2.7.3	Zhrnutie	23
2.8	FC Portugal 2001	23
2.8.1	Stratégia	24
2.8.2	Rozmiestňovanie	24
2.8.3	Výmena rolí	24
2.8.4	Pozorovací mechanizmus.....	24
2.9	Výber hráča.....	24
3	Špecifikácia	25
3.1	Neurónové siete	25
3.1.1	Použitie viacerých neurónových sietí	25
3.1.2	Neurónové siete naučené na základe rôznych súperových formácií	25
3.2	Pridanie akcií	26
3.3	Libero	27
3.4	Štatistická analýza rozloženia hráčov súpera	28
4	Hrubý návrh.....	29
4.1	Neurónové siete	29
4.1.1	Výber na základe informácií od Kouča.....	29
4.1.2	Výber na základe rozhodnutia sa hráča.....	29
4.2	Porovnávanie pomocou vzorou	31
4.3	Akcia	32
4.4	Generovanie vzorov a akcií.....	33
4.5	Libero – rozpoznanie situácie	33
4.5.1	Presun obranného libera do obrany	33
4.5.2	Presun obranného libera naspäť do stredovej formácie	34
4.5.3	Presun útočného libera do útoku	34
4.5.4	Presun útočného libera naspäť do stredovej formácie.....	35
4.6	Štatistická analýza rozloženia hráčov súpera	35
4.7	Prihrávky obsadeným hráčom	35
5	Prototyp	37
5.1	Použitie neurónových sietí pri určovaní pravdepodobností prihrávok	37
5.1.1	Súčasný stav.....	37

5.1.2	Opis implementácie algoritmu na určovanie prihrávkov	38
5.1.3	Opis implementovanej neurónovej siete	39
5.2	Prihrávky	40
5.2.1	Zisťovanie úspešnosti prihrávkov	40
5.2.2	Prihrávky do behu a obsadeným hráčom	43
5.3	Kouč	44
5.4	Vizualizácia rozhodovacieho stromu	44
5.5	Libero	45
5.6	Overenie použitia vzorov a masiek	45
5.7	Zhodnotenie prototypu	46
6	Opis riešenia	49
6.1	Úprava algoritmu na prihrávanie	49
6.2	Okolie ohrozenia spoluhráča	52
6.3	Nakopávaný dribblig	52
6.4	Problémy s kopaním	54
6.5	Trénovanie prihrávkov	55
6.5.1	Dôvody tréningu	55
6.5.2	Tréning	56
6.5.3	Spracovanie tréningových dát	59
6.6	Príčina pádu kouča	60
6.7	Najdôležitejšie zmenené a doplnené funkcie	61
6.8	Chyby v hráčovi	62
7	Zhodnotenie	63
7.1	Overenie riešenia	63
7.2	Čo sme nestihli	63
7.3	Čo sme sa naučili	63
7.4	Záver	64

Použitá literatúra

1

ZOZNAM TABULIEK

Tab. 2-1 Tím Squirrel Squadron: Problémy so správou „Pusti“	12
Tab. 2-2 Manažment energie hráča tímu L. A. S. T. United	16

ZOZNAM OBRÁZKOV

Obr. 2-1 Vnímanie sveta hráčom	4
Obr. 2-2 Tím FC Farmári- Matica rozmiestnenia hráčov (šedé políčka sú ohodnotené hodnotou 1)	8
Obr. 2-3 Tím FC Farmári – Parametre pri rozhodovaní o úspešnosti prihrávky	11
Obr. 2-4 Rozpad koordinačného grafu na skupinu podgrafov a nahradenie agentov rolami hráčov	23
Obr. 4-1 Zobrazenie pravdepodobností výberu jednotlivých neurónových sietí	30
Obr. 4-2 Výber výstupu konkrétnej neurónovej siete na základe určenia pravdepodobnosti rozhodovacou sieťou	31
Obr. 4-3 Umiestnenie hráča s loptou v mriežke	32
Obr. 4-4 Hierarchické akcie	33
Obr. 4-5 Prihrávka obsadenému hráčovi	36
Obr. 5-1 Histogram vyhodnocovania úspešnosti prihrávok neurónovou sieťou	42
Obr. 6-1 Situácia spoluhráč a dvaja protihráči	50

1 Úvod

Predstavte si...

...stojíte na futbalovom ihrisku preplnenom športovými fanúšikmi, chystáte sa na posledný výkop. Stav je nerozhodný. Oproti stojí súper, vyzerá ako obyčajný hráč, správa sa ako obyčajný hráč, má rovnaké fyzické možnosti ako bežný hráč, ale je niečím výnimočný. Je výsledkom polstoročia výskumu a tréningu, je to humanoidný robot, ktorý prvý krát stojí tvárou v tvár človeku a jeho jediným cieľom je vyhrať.

Nereálna predstava? Možno pre niekoho a v nejakom čase. Ale náš tím je tu preto, aby aj svojou troškou prispel k naplneniu tohto sna. Naším cieľom nie je vytvorenie fyzického robota schopného hrať, to prenecháme na skúsenejších zariadení z daného oboru. Nás zaujíma, ako sa budú títo hráči správať.

A čo to má spoločné s naším štúdiom? Takýto futbalový tím je multiagentový systém. Cieľom nášho projektu je vytvoriť, resp. vylepšiť, už existujúceho agenta. Našou snahou je posunúť úroveň hráčov vyvíjaných na FIIT STU opäť o krok dopredu, rozšíriť ich schopnosti a zmerať si sily v súťaži RoboCup.

1.1 O DOKUMENTE

Tento dokument bol vytvorený v rámci predmetu Tvorba softvérového systému v tíme. Obsah predstavuje riešenie zadania súvisiaceho s problematikou simulovaného robotického futbalu, ktorého liga nesie názov RoboCup (Robot World Cup Soccer Games and Conferences). Regionálne kolo sa uskutoční aj tento rok na pôde FIIT STU v Bratislave. Cieľom súťaže je neustále vylepšovanie možností hráčov zahrňujúcich novšie a lepšie stratégie, plánovania, spôsoby tréningu, komunikácie. Hra má presne určené svoje pravidlá, ktoré simulujú reálne podmienky hry.

Práve čítate prvú kapitolu, ktorá obsahuje stručný úvod do problematiky robotického futbalu a znenie zadania projektu.

Druhá kapitola obsahuje analýzu pravidiel hry, ktoré sú určené verziou herného servera Soccer Server. Určuje podmienky komunikácie, vlastnosti a ohraničenia hráčov vytvorených tímami študentov na fakulte počas predchádzajúcich rokov a tiež hráčov významných svetových tímov. Výstupom analýzy je porovnanie hráčov jednotlivých tímov a následný výber hráča, ktorého sme sa rozhodli použiť a vylepšiť jeho schopnosti.

Tretiu kapitoly tvorí špecifikácia možných vylepšení hráča a v štvrtej opisujeme algoritmy a postupy, ktoré plánujeme uplatniť počas implementácie.

1.2 ZADANIE

RoboCup – nové stratégie

Téme RoboCup, presnejšie lige simulovaného robotického futbalu sa naši študenti venujú už šesť 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.

V rámci fakulty sme realizovali viacero súťaží a posledné ročníky už boli oficiálnymi turnajmi 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 aj 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 spresnenie 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.

2 ANALÝZA

Táto časť obsahuje analýzu herného servera Soccer Server (kapitola 2.1), analýzu hráčov vytvorených tímami študentov na našej fakulte v predmete Tímový projekt za predchádzajúce roky (kapitoly 2.2 až 2.5) a analýzu hráčov niekoľkých významných zahraničných tímov (kapitoly 2.6 až 2.8). Výstupom analýzy je porovnanie hráčov jednotlivých fakultných tímov a následný výber hráča, ktorého sme sa rozhodli použiť a vylepšiť jeho schopnosti (kapitola 2.9).

2.1 SOCCERSERVER

2.1.1 Základné pojmy

Soccerserver je prostredie, ktoré poskytuje virtuálne hracie pole klientom predstavujúcich jednotlivých hráčov. Simuluje pohyb lopty a hráčov a kontroluje pravidlá hry. Komunikácia medzi serverom a klientmi je zabezpečovaná pomocou protokolu UDP.

Klient (hráč) môže byť vytvorený v ľubovoľnom programovacom jazyku, ktorý podporuje komunikáciu pomocou UDP rozhrania.

Soccermonitor je doplnkový program k Soccerserver. Je to rozhranie, ktoré pomáha zobrazovať hracie pole, vytvárané Soccerserver-om.

2.1.2 Analýza servera

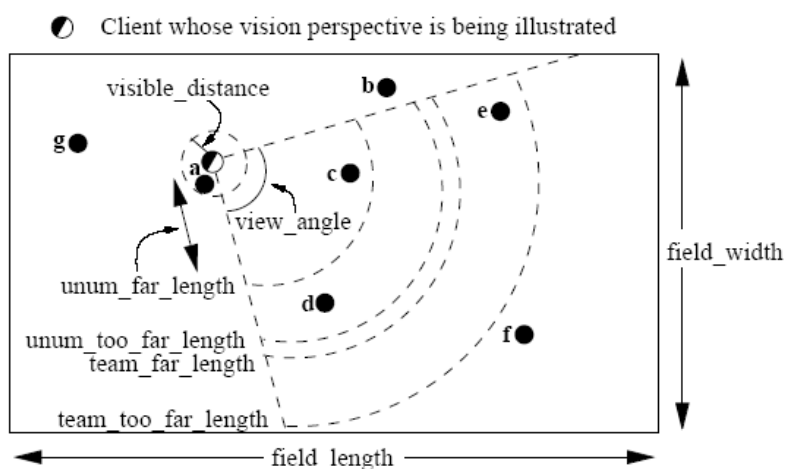
Server poskytuje hráčom senzorické informácie (o objektoch na hracom poli) a informácie o stave hry. Hráči posielajú serveru svoje požiadavky na akcie (kopni loptu, bež, otoč sa,..). Spracovanie dát serverom sa deje v reálnom čase, ale samotná hra (simulácia) sa odohráva v diskretných intervaloch.

Senzory hráčov

Každý hráč má tri rozličné senzory:

- *Sluchový senzor* – slúži na počúvanie správ od iných hráčov, od kauča alebo rozhodcu. Samotné počúvanie je realizované funkciou (*hear (Time) (Direction) (Message)*), kde Time je aktuálny čas, Direction je smer zdroja (ak je zdroj samotný klient, smer má hodnotu self, ak je zdroj rozhodca, potom má hodnotu referee). Ak chce hráč poslať správu ostatným hráčom, je mu to umožnené prostredníctvom funkcie (*send (Message)*). Táto správa nesmie byť dlhšia ako 10B.
- *Zrakový senzor* – slúži na vnímanie vizuálnej informácie o hracom poli. Informácie sú posielané pomocou funkcie (*see (Time) (ObjInfo) (ObjInfo) ..*) ObjectInfo je objekt, ktorý obsahuje napríklad informácie o relatívnej vzdialenosti (Distance), smere (Direction), natočení (FaceDir), čísle hráča

(UNum) a mene tímu (TeamName) objektu. S rastúcou vzdialenosťou objektu je doručované množstvo informácií obmedzované. Teda od určitej vzdialenosti objektu správa neobsahuje číslo hráča UNum. Ak je objekt príliš vzdialený správa tiež neobsahuje TeamName (príslušnosť k tímu). To isté platí o parametre Direction. Informácia je posiadaná hráčovi periodicky v pevne stanovenom časovom kroku. Kvalitu pohľadu možno z pohľadu hráča zmeniť a to pomocou funkcie (*change_view (ANGLE_WIDTH) (QUALITY)*), kde ANGLE_WIDTH musí byť jedna z hodnôt wide = 180°, normal = 90°, resp. narrow = 45°. QUALITY musí byť buď low (je posiadaná iba informácia o smere) alebo high (klient dostane informáciu aj o smere aj o vzdialenosti). Implicitné hodnoty parametrov sú normal a high. Hráč vníma aj virtuálne značky na hracom poli (bránka, rohy, stred ihriska a ďalšie). Hráč čiastočne vidí aj objekty, ktoré nie sú v jeho zornom uhle (a sú blízko) – v tomto prípade vníma len typ objektu, ale nie jeho exaktné meno.



Obr. 2-1 Vnímanie sveta hráčom

- Telový senzor** – umožňuje hráčovi zistiť svoj fyzický stav pomocou funkcie (*sense_body (TIME) (view_mode QUALITY WIDTH) (stamina STAMINA EFFORT) (speed AMMOUNT_OF_SPEED) (kick KICK_COUNT) (dash DASH_COUNT) (turn TURN_COUNT) (say SAY_COUNT)*) – Tu sú určené všetky parametre hráča ako napríklad energia, rýchlosť, natočenie tela, hlavy. Informácia je posiadaná periodicky v pevne stanovenom časovom kroku.

Módy hry a riadiace príkazy

Módy hry sú nasledovné

Play_on

Je "hrací" mód, pri ktorom je lopta v pohybe. Nastane ak sa lopta začne pohybovať vďaka príkazu kick z ľubovoľného iného hracieho módu (kick_off, throw_in, goal_kick, corner_kick).

Kick_off

Rozohranie lopty, pri ktorom musí byť každý hráč na svojej strane. Ak sa na nej nenachádza, simulátor ho umiestni na náhodné miesto na príslušnej polovici hracieho poľa.

Presunutie

Nastáva pri rozohrávaní lopty (*kick_off*, *throw_in*, *corner_kick*, *goal_kick*, *offside*). Simulátor presunie hráčov tak, aby boli minimálne vo vzdialenosti 9.15 od lopty. Hráči sú presunutý na obvod pomyselných kružníc

Gól

Nastane v momente, keď sa lopta dostane celým objemom do brány. Simulátor preruší zápas na 5 sekúnd, presunie loptu do stredu ihriska, pošle správu všetkým hráčom. Počas prerušenia sa hráči môžu presunúť na svoju polovicu pomocou príkazu 'move'. V opačnom prípade ich presunie simulátor a umiestni ich na náhodnej pozícii. Simulátor zmení hrací mód na *kick_off*.

Aut

Nastane, keď sa lopta dostane mimo hracieho poľa. Simulátor presunie loptu na príslušnú pozíciu (roh, aut, bránková čiara) a zmení hrací mód na *kick_in*, *corner_kick*, *goal_kick*.

Polčas

Simulátor preruší zápas v polovici plánovaného času trvania. Implicitná dĺžka polčasu je 3000 simulačných cyklov (okolo 5. min). Pokiaľ je po skončení zápasu skóre nerozhodné, predlžuje sa až kým sa nedosiahne gól – tzv. rýchla smrť.

V nasledujúcom texte sú popísané najdôležitejšie riadiace funkcie:

move(X Y)

Presun hráča na pozíciu (X,Y). Stred súradnicovej sústavy je v centrálnom kruhu, kladný smer osi X je určený smerom k súperovej bráne, kladný smer osi Y je určený smerom k pravej čiare (ktorá je určená vzhľadom na kladný smer osi X). Tento príkaz je k dispozícii iba v móde *before_kick_off*.

dash Power

Zrýchlenie hráča v smere jeho orientácie Z o hodnotu Power. Power môže byť z intervalu -30 až 100

kick Power Direction

Kopnutie do lopty silou Power a so smerom Direction. Možno iba ak je hráč bližšie ako *kickable_margin + ball_size + player_size*. Power môže byť z intervalu -180 až 180 a Direction z intervalu -180 až 180.

catch Direction

Príkaz môže použiť iba brankár a slúži na chytenie lopty v smere Direction. Loptu môže chytiť iba vtedy, ak sa nachádza v obdĺžniku s rozmermi *goalie_catchable_area_w* x *goalie_catchable_area_h* a v smere Direction. Ak chytenie nebolo úspešné, určitý čas nie je možné akciu chytania lopty znovu opakovať.

Heterogénni hráči

Soccerserver umožňuje hru s heterogénnymi hráčmi. Na začiatku hry vygeneruje zo základného hráča ďalších 6 náhodných typov. Heterogénni hráči majú niektoré vlastnosti oproti základnému hráčovi zmenené (hráč môže byť rýchlejší, presnejší, silnejší v kope... ale i naopak). Reálne to znamená, že heterotyp môže byť výborný na niektoré špecifické herné činnosti a pozície a na iné je absolútne nevhodný.

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.

Kouč a Tréner

Rozdiel medzi koučom a trénerom je nasledovný. Kouč je súčasťou hry aj pri riadnom zápase a interaguje s hráčmi pomocou posielania zvukových správ. Tréner môže byť použitý iba v tréningových zápasoch (nie počas riadnych zápasov).

Kouč dostáva na rozdiel od hráčov nezašumené informácie o dianí na ihrisku. Jeho hlavnou úlohou je sledovať a analyzovať hru. Na základe získaných informácií môže kouč meniť stratégiu hry (napr. zmena formácie), striedať hráčov alebo posilať dôležité pokyny. Komunikácia s hráčmi je však obmedzená na maximálny počet odoslaných správ. Pre komunikáciu medzi koučom a hráčmi bol vyvinutý špeciálny jazyk CLang.

2.1.3 Prehľad verzií SocckerServer

Verzia servera 7.X

- Bola pridaná podpora nového kouč jazyka – CLang a ďalšie rozšírenie možností kouča.
- Pri výkope pred bránky (goal kick) musí lopta opustiť pokutové územie. Pred opustením pokutového územia sa jej nesmie dotknúť žiadny hráč.
- Bol pridaný typ hráča „Heterogenous player“ – je možné zmeniť parametre hráča v určitých medziach. Takto je možné špecializovať hráčov napríklad ako obrancov a útočníkov.
- Hráč nemôže rozohrať loptu sám sebe. Ak tak urobí, loptu získa druhý tím.
- Hráč nemôže nahrávať vlastnému brankárovi.

Verzia servera 8.X

- Možnosť zobrazit' logo tímu vo formáte XPM. Logo nastavuje kouč.
- Sprísnené pravidlá hry podľa pravidiel FIFA.
- Zmena formátu „hear“ správy.
- Server podporuje kompresiu záznamov stretnutí a komunikácie medzi hráčmi.

- Hráč môže pri prihlásení na server (init) ohlásiť verziu servera, nad ktorou pracuje a správy mu budú potom posielané v tom formáte.
- Hráč môže prednostne prijímať správy od konkrétneho hráča pomocou príkazu (attentionto). Hráč môže ignorovať niektoré správy (hear) pomocou príkazu (ear).
- Hráči môžu podľa nových pravidiel počuť jednu správu od každého tímu za cyklus, maximálna dĺžka správy je desať bajtov. Správy (hear) od servera chodia vždy až v ďalšom cykle po správe (sensebody).
- Zmenený formát správy (hear) a príkazu (say).
- Hráč môže vypichnúť loptu pomocou príkazu (tackle).

Verzia servera 9.X

- Strelenie gólu je teraz posudzované podľa pravidiel FIFA. Pre uznanie gólu musí byť lopta úplne za čiarou
- Brankár môže použiť (catch) príkaz hocikde na ihrisku, ale ak chytí loptu mimo 16-čky, považuje sa to za ruku a nasleduje výkop pre protihráča na mieste, kde brankár chytil loptu.
- Pokiaľ sa brankár ocitne s loptou v bráne, považuje sa to za gól.
- Pokiaľ sa brankár ocitne s loptou mimo ihriska považuje sa to za aut alebo roh.
- Zmenená dynamika brankára.

Verzia servera 10.x

- Najnovšia verzia servera, ktorá je zatiaľ funkčná iba pod operačnými systémami typu UNIX.
- Nepodarilo sa nám zistiť informácie o zásadnejších zmenách oproti ostatným verziám.

2.2 FC FARMÁRI

2.2.1 Základné informácie

FC Farmári je minuloročný tím (t.j. šk. rok 2004/2005). Hráč nadväzuje na hráča tímu Stjupit Dox (z roku 2002/2003), ktorého vylepšuje a opravuje niekoľko jeho chýb. Na turnaji RoboCup at FIIT'05 tím FC Farmári suverénne zvíťazil.

FC Farmári sa zamerali na riešenie problémov z nasledujúcich oblastí:

- Analyzátor ihriska v koučovi, na základe obsadenia častí ihriska súperom
- Tvorba a vyhodnotenie pokrytia subformácií s použitím neurónových sietí
- Neurónová sieť učiaca sa driblovať s loptou
- Výber spoluhráča na prihrávku na báze učenia s odmenou

- Úprava brankára

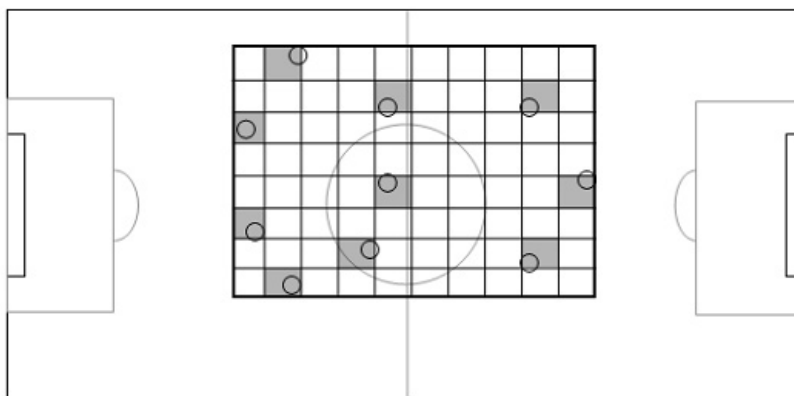
2.2.2 Analyzátor ihriska v koučovi

Hlavnou úlohou implementovaného kouča je sledovanie zápasu a navrhovanie vhodných zmien formácií alebo iných prvkov, ktoré by mohli viesť k víťazstvu tímu. Kouč tak robí na základe analýzy obsadenosti častí ihriska súperom.

Rozdelenie ihriska

FC farmári rozdelili celé ihrisko na sektory o rozmeroch 2x2 metre. Štatistika obsadenosti jednotlivých sektorov ihriska sa uchováva v dvojrozmernom poli celých čísel. Do tohto poľa sa každý cyklus pre jednotlivé sektory vkladá počet protihráčov, ktorí sa v danom cykle nachádzajú v jednotlivých sektoroch. Štatistika sa vyhodnocuje každých 500 cyklov. Na jej základe sa potom vykoná úprava formácie tímu nasledujúcim spôsobom:

- Aby boli brániaci hráči vo formácii, pri ktorej sú čo možno najčastejšie v blízkosti sektorov často obsadených protihráčmi.
- Aby boli útočníci hráči vo formácii, pri ktorej sú čo najčastejšie v blízkosti sektorov najmenej obsadených protihráčmi.



Obr. 2-2 Tím FC Farmári- Matica rozmiestnenia hráčov (šedé políčka sú ohodnotené hodnotou 1)

Analýza súperových formácií

Dôvodom na analýzu súperových formácií je prispôbovanie sa hre súperovi, zistenie formácie, v ktorej sú jeho hráči usporiadaní. Na realizáciu analýzy formácie súperovho tímu je použitá dopredná neurónová sieť. Jej vstupom sú informácie o obsadenosti sektorov na ihrisku. Jej výstupom, je jedna z 20 definovaných formácií. Keďže spoľahlivosť siete pri klasifikácii daných formácií nie je 100 %, kouč počas hry každých 5 cyklov nechá klasifikovať danú formáciu neurónovou sieťou. Zapamätá si celkovo 10 posledných klasifikovaných formácií, ktorým pridelí váhy na základe ich aktuálnosti (najmenšiu váhu bude mať najstaršia formácia, a naopak). Potom určí formáciu, ktorá má najvyšší koeficient (na základe váhy a početnosti).

Skladanie formácií

Pre účely adaptácie formácií súperovej hre bude mať kouč k dispozícii rôzne subformácie, z ktorých poskladá výslednú formáciu. Výhodou subformácii je, že takýmto spôsobom sa dá pokryť podstatná časť formácií. Čiže súper má sťaženú úlohu klasifikovania danej formácie.

Základné typy formácií (počet obrancov - počet stredopoliarov - počet útočníkov):

- 4-4-2 – obranná, udržanie skóre
- 4-2-4 – útočná
- 3-4-3 – útočná, silný stred poľa
- 4-3-3 – alternatíva k formácii 3-4-3 s posilnením obrany
- 3-3-4 – alternatíva k formácii 3-4-3 s posilnením útoku
- 3-2-5 – formácia na extrémny útok

Subformácia je modifikovanie formácie v malom rozsahu tak, že sa zmení rozostavenie hráčov v jednotlivých častiach danej formácie.

Heterogénni hráči a striedanie

Kouč je schopný využiť striedanie počas zápasu na základe hore uvedenej analýzy zápasu. Cieľom je nájsť hráčov, ktorých vlastnosti sú vhodné pre ich funkciu v rámci formácie, čo sa dosiahne definovaným spôsobom klasifikácie a zoradením určitých vlastností hráča na základe ich dôležitosti pre danú úlohu vo formácii.

2.2.3 Tvorba a vyhodnotenie pokrytia subformácií

Na tvorbu subformácií vytvoril tím FC Farmári aplikáciu UnitCreator. Je to editor, ktorý má zjednodušiť a zautomatizovať prácu používateľa pri definovaní nových subformácií. Používateľ označí pozície jednotlivých hráčov pre danú formáciu a aplikácia vygeneruje výstupný súbor, v ktorom bude subformácia uložená v textovej forme.

Pre danú subformáciu sú definované nasledujúce hodnoty:

- Name – názov formácie, ktorej súčasťou je daná subformácia
- Width a Height – rozmery subformácie v metroch
- Scale – dĺžka resp. šírka štvorcového sektora (vytvárajú celú subformáciu) v metroch
- Hráč je definovaný parametrami:
 - X-ová pozícia, Y-ová pozícia – pozícia hráča v subformácii v metroch vzhľadom na stred formácie.
- Range – vzdialenosť, ktorú dokáže hráč dosiahnuť za maximálny povolený čas.

2.2.4 Neurónová sieť učiaca sa driblovať s loptou

Úlohu zabezpečoval tréner tímu Stjupit Dox, ale boli potrebné zmeny. Na učenie driblovania s loptou bola použitá 3-vrstvová neurónová sieť s učením pomocou backpropagation. Neuróny siete používajú ako aktivačnú funkciu unipolárnu sigmoidu:

Vstupy siete sú:

- Relatívny vektor lopty, čiže dve súradnice vektora lopty vzhľadom na hráča.
- Relatívny vektor cieľa, čiže dve súradnice vektora cieľa vzhľadom na hráča.
- Akcia: tri druhy akcií (Kick, Turn, Dash).
- Parametre akcií
- Vzdialenosť lopty od cieľa.
- Vzdialenosť hráča od cieľa.
- Ostávajúci čas, čiže čas ktorý nám zostáva na dodriblovanie do cieľa.

Vektory sú reprezentované ako reálne čísla. Učenie prebieha metódou odmeny a trestu. V prípade úspechu je hodnota výstupu blízka alebo rovná 1 a v prípade neúspechu blízka alebo rovná 0.

Samotné tréningovanie je nasledovné:

Tréner nastaví hráčovi cieľ pevne na stred ihriska a umiestni ho náhodne na nejakú inú časť ihriska. Taktiež mu pevne stanoví aj informáciu o počte cyklov, za ktoré má k danému cieľu dodriblovať (50 cyklov). Ak sa hráčovi podarí dodriblovať loptu do cieľa v stanovenom čase, je zaznamenaný úspech.

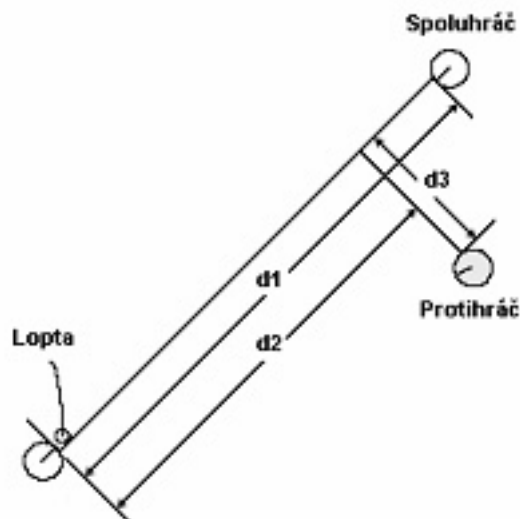
Poznámka: Pri reálnom zápase určuje cieľ taktická vrstva.

2.2.5 Výber spoluhráča na prihrávku na báze učenia s odmenou

Túto časť navrhol a implementoval tím FC Farmári neskôr a jej cieľom bolo naučiť hráča čo možno najlepšie sa rozhodnúť o realizácii prihrávky. Algoritmus sa opiera o analytický výpočet vzdialeností potenciálneho adresáta prihrávky a jeho protihráčov. Na tento cieľ tím implementoval aplikáciu HotParser, ktorý zo vstupných dát, získaných počas reálnych zápasov, vygeneruje dáta v textovom formáte pre učenie sa neurónovej siete. Vstupný súbor obsahuje zoznam špecifických udalostí (lopta kopnutá, lopta chytená), ktoré nastali počas zápasu. Na základe úspechu - neúspechu sa daná neurónová sieť učí, tj. či sa pri daných parametroch prihrávky oplatilo túto prihrávku realizovať. Po natrénovaní hráča sa na základe uvedených parametrov pomocou neurónovej siete rozhodne, či prihrávka prejde k spoluhráčovi alebo nie. Hráč sa rozhodne na základe pravdepodobnosti úspešnosti prihrávky, ktorú mu určí neurónová sieť.

Vstup pre každú prihrávku je daný na obrázku, pričom

- d_1 je vzdialenosť od potenciálneho spoluhráča na prihrávku
- d_2 je vzdialenosť od protihráča v smere prihrávky
- d_3 je vzdialenosť od protihráča v smere kolmom na prihrávku



Obr. 2-3 Tím FC Farmári – Parametre pri rozhodovaní o úspešnosti prihrávky

2.2.6 Úprava brankára tímu Stjupit dox

Úprava sa týkala najmä schopnosti brankára odhaľovať heterotypy protihráčov a vhodne na ne reagovať. Preto tím FC Farmári upravili pôvodného brankára tak, aby nepočítal úspešnosť chytienia lopty z konštanty rýchlosti hráčov, ale aby si zistil rýchlosť možných okolitých protihráčov (na základe ich heterotypu) a z nich zistil svoje možnosti loptu zachytiť.

2.2.7 Zhodnotenie

V prípade systému analyzátora formácií boli dosiahnuté výsledky tímom FC Farmári hodnotené pozitívne. Použitie neurónových sietí pri driblingu bolo neúspešné, čo môže súvisieť s nevhodnou sadou vstupov neurónovej siete alebo s použitím reálnych čísel ako vstupmi siete. V prípade použitia neurónovej siete na určovanie úspešnosti prihrávky, sú výsledky hodnotené pozitívne. Úprava pôvodného brankára bola taktiež hodnotená ako úspešná. Tím sa umiestnil na turnaji RoboCup at FIIT'05 na prvom mieste, s jednou prehrou v základnej skupine a s dvoma remízami v finálovej skupine.

2.3 SQUIRREL SQUADRON

2.3.1 Základné informácie

Squirrel Squadron je vlnajší tím (t.j. 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 zamerl na riešenie problémov nasledujúcich oblastí:

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

2.3.2 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 tebou, prihraj, chytaj, mám, bež, pusti, správy na riadenie ofsajdovej pasce, pokry, AttentionTo a správy na komunikáciu s koučom. Podrobnejšie sa tým zamerlal na správy pokry a pusti.

Správa „pokry“

Slúži na zabránenie krížnych prihrávok 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.

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 nastaví príznak „nebež za loptou“. Implementácia tohto vylepšenia priniesla so sebou niekoľko problémov, ktoré sumarizujem aj s príslušným riešením v nasledujúcej tabuľke:

Problém	Opis problému	Riešenie
Znefunkčnenie prihrávok	Prihrávajúci hráč v nasledujúcom cykle zistí, že je najbližšie k lopte a kričí správu Pusti.	Hráč kričí správu Pusti len v prípade, že sa lopta nepohybuje smerom od neho príliš rýchlo.
Kedy zrušiť príznak „Nebež za loptou“?	Príznak si hráč vynuluje, keď zistí, že loptu drží iný hráč. Tento spôsob nefunguje, keď sa lopty niekto dotkne v momente, keď daný hráč na loptu nevidí.	Hráč nenuluje príznak vtedy, keď vidí, že sa niekto iný dotkne lopty, ale vtedy, keď lopta výrazne zmení vektor svojho pohybu.
Deadlock hráčov	Viacerí hráči si myslia, že sú najbližšie k lopte a zakričia správu Pusti.	Hráč spolu so správou Pusti kričí aj svoju odhadovanú vzdialenosť od lopty. Ak jeden hráč, aj vyšle, aj prijme správu Pusti, porovná svoju vzdialenosť s prijatou vzdialenosťou a na základe toho sa rozhodne, či pobeží ďalej za loptou alebo ju prenechá spoluhráčovi.

Tab. 2-1 Tím Squirrel Squadron: Problémy so správou „Pusti“

2.3.3 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.

Využitie štatistík

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

- výskyt lopty v určitých zónach ihriska
- poloha spoluhráčov, resp. protihráčov, v zónach
- kontrolovanie lopty

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.

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.

2.3.4 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)

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 prihrávkou. 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.

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 L. A. S. T. UNITED

2.4.1 Základné údaje

Tím L.A.S.T. United pôsobil na FIIT STU v akademickom roku 2003/2004. Tento tím sa pokúsil vytvoriť nového hráča prakticky od základov, čo sa odrazilo na výsledku turnaja, keď utrpeli tri prehry a jednu remízu a nepostúpili do finále.

Hráč je implementovaný pomocou skriptovacieho jazyka Lua, ktorý bol podľa nich vhodným prostriedkom pre súčasné použitie procedurálneho, funkcionálneho a objektového programovania. Výhodou je rýchly preklad a možnosť ladenia hráča počas simulácie, keďže odpadol čas potrebný na kompiláciu. Nevýhodou sú nároky na systémové prostriedky.

Návrh architektúry hráča bol viac krát upravovaný. Pôvodný návrh počítal s tým, že v jazyku Lua budú implementované vrstvy zodpovedné za plánovanie, komunikáciu hráčov a reaktívnu logiku. V jazyku C++ to budú základné schopnosti hráča, vnútorný svet hráča a fyzická komunikácia so serverom. Nakoniec sa tím rozhodol pre kompletnú implementáciu pomocou jazyka Lua, čím sa odstránili problémy s rozhraním medzi jazykmi.

Autori chceli striktne základné schopnosti hráča, ku ktorým patrí napr. kopnutie do lopty, oddeliť od reaktívnej logiky. Následkom toho sa hráč snažil napr. kopnúť do lopty, aj keď to nebolo možné.

2.4.2 Komunikácia

Pôvodným zámerom tímu bolo použiť zvukovú komunikáciu na plánovanie formácií hráčov pomocou hráča, ktorý bol aktuálne pri lopte.

Bol navrhnutý jeden hráč, tzv. Spojár, ktorý počúval hráča s loptou, tzv. Veliteľ, a tieto správy kričal ostatným, ktorí počúvali iba jeho. Spojárom mal byť stredopoliar, ktorý hrá v strede ihriska, vďaka čomu ho počujú zvyčajne všetci hráči (dosah 50 m). Veliteľ a si určoval Spojár, problém nastal vtedy, ak nevidel na hráča a nemohol identifikovať jeho číslo.

Veliteľ nemal informáciu, či je veliteľom, mohol to len usudzovať podľa vzdialenosti od lopty. Keďže bol najbližšie k lopte, mohol najlepšie ohodnotiť situáciu a vybrať nasledujúce akcie a formácie pre daný aktuálny stav.

Ostatní hráči môžu v každom cykle kričať svoj vnútorný svet, keďže správa od spojára sa doručí prednostne. Spojár nemôže kričať svoj vnútorný stav.

Spojára vyberá kauč, ktorý to oznámi všetkým hráčom.

Na túto komunikáciu sa využíval príkaz AttentionTo.

Od plánovania nakoniec tím odstúpil pre nedostatok času. Namiesto toho použil dynamické formácie

Správa obsahuje nasledujúce hodnoty:

- ID hráča (od koho je správa)
- ID hráča (pre koho je správa)
- ID formácie
- pozícia lopty, kde sa snaží hráč dostať loptu
- energia hráča, ktorý posielal správu

Vďaka informácii o energii hráča je možné naplánovať napr. spätnú prihrávku.

2.4.3 Rozoznávanie situácie na ihrisku

Na tento účel autori navrhovali použiť kvadrantové stromy na rozoznávanie obrazov. Rozpoznávaným obrazom je aktuálna situácia na ihrisku, rozmiestnenie hráčov, poloha lopty...

2.4.4 Strategické rozmiestňovanie

Pod týmto pojmom rozumeli zápis znalostí členov tímu o futbale. Rozdelenie hráčov je dané formáciami. Každému hráčovi vo formácii sa priradí rola. Rola opisuje pre hráča útočnú, obrannú pozíciu a pozíciu v strede ihriska.

Vďaka tomu, že všetci hráči sa správajú podľa danej formácie, vedia si odhadnúť polohu spoluhráčov, ak o nich nemajú aktuálnu informáciu.

Pre danú rolu je definované, ktorých spoluhráčov má sledovať, komu má prihrať, ktorým smerom strieľať.

Zmenu formácie môže prikázať Veliteľ alebo Spojár pomocou komunikácie, čo je vhodné najmä pri útoku, kedy je žiaduce zmeniť správanie sa hráčov.

2.4.5 Manažment energie hráča

Riadenie spotreby hráča závisí od aktuálnej situácie na ihrisku. Autori zdefinovali tri ľahko rozoznateľné situácie a k nim prislúchajúce správanie sa:

Situácia	Správanie
lopta za hranicou šestnástky	bež za loptou na 75%
lopta je blízko hráča	bež za loptou na 100%
ostatné situácie	nebež za loptou

Tab. 2-2 Manažment energie hráča tímu L. A. S. T. United

Obmedzenia platia pre prípad, že by energia klesla pod hranicu 1200 jednotiek.

2.4.6 Skenovanie situácie

Autori navrhovali zaviesť štyri typy sledovaných objektov, podľa priority: prioritný bod, lopta, zoznam bodov, okolie. Prioritný bod sa nastavuje vyššou logikou a sleduje sa určitý počet krokov. Loptu hráč sleduje, ak sa nachádza v stanovenej vzdialenosti, inak s ostatnými objektmi.

2.5 SKLO

Tím Sklo je výsledkom práce študentov z predmetu Tímový Projekt inžinierskeho štúdia na FEI STU z akademického roku 2003/2004. Autori pri tvorbe tímu vychádzali z tímu Stjupit Dox, ktorý vznikol rovnakým spôsobom v školskom roku 2002/2003. Pokúsili sa vylepšiť inteligenciu hráčov pridaním, resp. vylepšením nasledujúcich vlastností:

- rozšírenie a prepracovanie vnútorného sveta hráča
- zdokonalenie komunikácie hráčov
- pridanie dvoch funkcií ovplyvňujúcich taktiku hráča
- plánovaný výkop hráča

- diferenciácia taktiky podľa typu hráča
- rozlišovanie heterotypov súpera koučom

2.5.1 Vnútorný svet hráča

Vytvorením triedy PlayerArray sa odstránila dátová duplicita, keďže obsahuje okrem informácií o hráčovi aj funkcie, ktoré predtým existovali zvlášť pre spoluhráča aj protivráča. Navyše boli pridané aj funkcie na nájdenie hráča podľa rôznych kritérií, ako napríklad výber vhodného kandidáta na prihrávku.

Ďalej boli pridané polia pre heterotypy hráčov (vlastných aj súperov). Heterotypy súperov hráčom posielajú kouč prostredníctvom freeform message párkrát za zápas (nehrozí zahltenie).

Triedu Player nahradila trieda SimPlayer, ktorá ju rozširuje najmä o schopnosť predikcie počtu cyklov, za ktoré sa hráč dostane k určitému bodu alebo pohybujúcemu sa objektu. Implementácia sa nevyhla problémom, ktoré tvorí najmä veľká časová náročnosť.

Zaujímavé rozšírenie predstavuje aj objekt PrepareCommands, ktorý v sebe zahŕňa plánované akcie hráčov a metódy na prácu s nimi.

Kvôli vývoju zložitejších algoritmov boli vytvorené nasledujúce podporné triedy:

- Abscisse – úsečka. Obsahuje funkcie na zistenie, či bod leží na danej úsečke, prípadne dokáže vytvoriť rovnobežnú úsečku k danej úsečke.
- Circle – kružnica. Združuje metódy na zistenie, či sa bod nachádza vo vnútri kružnice, prípadne v ktorom kvadrante leží.
- Line – priamka. Prostriedok na zistenie vzťahov medzi priamkou a kružnicou, priamkou a bodom a dvomi priamkami.
- Ray – polpriamka. Dokáže nájsť priesečník polpriamky a kružnice.
- Log – slúži na zaznamenávanie informácií o hráčovi počas behu simulácie. Je vhodná najmä pri sledovaní veľkého množstva parametrov.

2.5.2 Zvuková komunikácia

Správy sa rozdelili na dva typy, tzv. bežné správy, kde hráč posielá základné informácie o svojom vnútornom svete len keď je na rade a urgentné správy, ktoré môže posielat' neustále.

2.5.3 Všeobecné taktické funkcie

Prvá funkcia predstavuje dynamické určovanie cieľa hráča driblujúceho s loptou. Zámer je driblovať smerom, kde je čo najmenej hráčov.

Druhá funkcia sa snaží nájsť vhodné miesto, kam kopnúť loptu, keď hráč nemôže ďalej bezpečne driblovať.

2.5.4 Plánovaný výkop

V prípade štandardných situácií, si hráč vytvorí vlastný plán, ako výkop vykoná. Plán je tvorený nasledovnými akciami:

- otočenie sa o 360 stupňov (preskúmaj okolie)
- pauza 10 simulačných krokov (kým sa nerozostavia spoluhráči)
- výber adresáta prihrávky
- odoslanie správy „chytaj“ s príslušnými parametrami
- otočenie sa na adresáta prihrávky a jej vykonanie

2.5.5 Taktika hráčov

Hráči sa delia na útočníkov, stredopoliarov, obrancov a brankára. Navyše sa rozlišuje či hráč má alebo nemá loptu.

Útočník s loptou najskôr zruší predchádzajúcu akciu a zakričí spoluhráčom správu „Mám“, aby ich upozornil, že má loptu. Ak sa nachádza na dostrel od brány alebo existuje určitá šanca na strelenie gólu, kopne hráč loptu do vypočítanej pozície v súperovej bránke. Inak sa hráč rozhoduje pre akcie v nasledujúcom poradí:

- prihrávka
- rýchle driblovanie
- bezpečné driblovanie
- núdzový kop

Správanie sa útočníka bez lopty bolo modifikované tak, aby hráč šetril čo najviac energiou. Šetrenie energiou sa aktivuje vtedy, keď nemá zmysel bežať naplno, to znamená v prípadoch, ak je hráč ďaleko od lopty a lopta smeruje od hráča. Ak je útočník bez lopty vo výhodnej pozícii, zakričí hráčovi s loptou, aby mu prihral. Ak nemá loptu súper, overí si hráč či nemá nastavený nejaký cieľ. Pokiaľ áno, beží k danému cieľu, pričom sa snaží pozeráť na loptu. Pokiaľ má loptu spoluhráč, uteká hráč na takú pozíciu, aby mu mohol spoluhráč výhodne nahráť. Ak sa ocitne v ofsajde, snaží sa z neho uniknúť. V prípade, že má loptu protihráč snaží sa útočník postaviť do takej pozície, aby mohol loptu zachytiť. Ak sa útočník vzdiali na viac ako 5m od svojej pozície v rámci formácie, vracia sa na miesto, pričom sa snaží mať loptu neustále na očiach.

Taktika stredopoliaru tímu Sklo je len mierne upravená oproti taktike stredopoliarov v tíme Stjupit Dox.

Hlavnou prioritou obrancu je zabrániť súperovým útočníkom strieľať na bránu. V prípade, že nemá loptu a je schopný ju získať, pokúsi sa o to. Ak má loptu súper a je mimo jeho dosahu, postaví sa medzi loptu a súpera, aby tak zabránil nahrávke. Do úvahy sa berie aj to, či už daný súper nie je pokrytý iným hráčom.

Ak má obranca loptu, snaží sa kopať na bránu, ak má šancu na úspech. Inak sa pokúsi nahráť spoluhráčovi dopredu alebo driblovať. Ako poslednú možnosť nahrá niekomu v rámci formácie.

Zmena brankára sa týka optimalizácie výkopu. Brankár vykonáva akcie podľa nasledujúcej postupnosti:

- analýza najvhodnejšieho miesta na výkop (10 cyklov)
- presun na vybrané miesto
- opakovanie analýzy na vybranom mieste (10 cyklov)
- natočenie a výkop
- správa „chytaj“

2.5.6 Rozlišovanie heterotypov koučom

Cieľom bolo naučiť kouča rozoznávať heterotypy súperov a následne získané informácie poslať hráčom

Rozoznávanie vychádza z predpokladu, že aspoň raz v zápase bude každý hráč nútený využiť svoje schopnosti na maximum, čím sa automaticky prezradí.

Kouč sleduje súperových hráčov a zaznamenáva maximá niektorých charakteristík. Za rozpoznanie heterotypu sa považuje moment, v ktorom hráč výrazne prekročí hraničnú hodnotu niektorej charakteristiky. Tréner sleduje nasledovné charakteristiky:

- rýchlosť pohybu hráča – keďže rýchlosť na konci cyklu je znížená o hodnotu `player_decay`, počíta sa ako rozdiel aktuálnej polohy a polohy hráča v predchádzajúcom cykle
- zrýchlenie
- rýchlosť otáčania hráča – vypočíta sa ako rozdiel aktuálneho uhla natočenia od uhla natočenia v predchádzajúcom cykle
- vzdialenosť, v ktorej hráč dokopne do lopty

Pri rozpoznaní heterotypov bol veľkým problémom šum, ktorý do simulácie vnáša server. Pomocou vyššie sledovaných charakteristík sa podarilo kouča naučiť rozpoznať len heterotypy na základe rýchlosti a zrýchlenia. Rozoznávanie na základe rýchlosti otáčania a vzdialenosti od lopty sa implementovať nepodarilo.

2.5.7 Zhmutie

Tímu Sklo sa podarilo zlepšiť rozohrávku od brány a obranu, vďaka čomu inkasujú veľmi zriedka. Úspech hry však spočíva v prehustení obranného pásma piatimi hráčmi, čím sa veľmi oslabil hra v strednom pásme a v útoku, takže väčšina zápasu sa odohráva v obrannom pásme.

Vďaka zdokonalenej komunikácii sa zlepšila rozohrávka hráčov a občas sa podarí prienik do súperovej obrany. Na druhej strane sa stáva pomerne často, že hráč nahráva loptu spoluhráčovi aj vtedy, ak sa medzi ním a spoluhráčom nachádza súper. Takto hráči zbytočne prichádzajú o loptu a tím aj o možnosť rozvinúť útočnú hru.

V dôsledku zamerania sa na obrannú hru je tím Sklo pomerne slabý v streľbe gólov. Veľká časť gólov navyše padá po chybe súperovho brankára, najčastejšie z brejkových situácií.

Väčšina zápasov tímu Sklo končila remízou alebo tesným víťazstvom, čo sa dá pokladať za úspech. Ak by sa tvorcovia popracovali aj na hre v strednom pásme a v útoku, mohol by tím skončiť výrazne lepšie.

2.6 BRAINSTORMERS

2.6.1 Základné informácie

Nemecký tím Brainstormers z univerzity Osnabrück je víťazom tohtoročného svetového šampionátu RoboCup-2005 a aj v predošlých ročníkoch sa umiestnil na popredných miestach (od roku 2000 pravidelne na 2. až 3. mieste). Web-stránka tímu je momentálne nedostupná, preto budem vychádzať len z minuloročných analýz iných tímov (FC Farmári, 2005; Squirrel Squadron, 2005).

2.6.2 Architektúra hráča

Pôvodne bola architektúra hráča delená do 3 vrstiev. Jej hlavnou nevýhodou bolo, že riadenie prebiehalo výlučne v smere vrstiev zhora nadol: Stratégia (Strategy) – Taktika (Policy) – Pohyby (Moves). Nižšia vrstva nemohla ovplyvniť vyššiu vrstvu, aj keď to bolo niekedy vhodné. Tiež bolo nutné každú vlastnosť hráča jednoznačne zaradiť do určitej vrstvy, čo nie je často možné.

Tím preto prebudoval architektúru hráča. Nová architektúra sa teraz zakladá na modeloch správania, pričom zachováva určitú hierarchiu, ale zároveň odstraňuje problémy predchádzajúcej verzie architektúry.

2.6.3 Individuálne stratégie hráčov

Správanie sa jedného hráča je možné chápať ako jednoagentový Markovovský rozhodovací proces. Potom hra celého tímu je výsledkom správania sa jednotlivých hráčov. Tímová stratégia je multiagentový Markovovský proces. Individuálne stratégie hráčov – agentov sú tréňované pomocou neurónovej siete s odmenou a trestom. Úspešné realizovanie zamýšľanej činnosti sa ohodnotí známku 1 – úspech, zlyhanie sa ohodnotí známku -1 – neúspech. Natréňovanú neurónovú sieť použil tím Brainstormers na určenie, ktorú akciu má hráč vykonať, aby mal čo najväčšiu šancu na úspech. Tréňované boli zvlášť situácie pre hráča s loptou a pre hráča bez lopty:

Možné akcie pre hráča s loptou sú:

- prihrávka priamo na vybraného spoluhráča
- prihrávka do priestoru, kde je veľká pravdepodobnosť, že spoluhráč sa dostane k lopte skôr než protivník
- dribling s loptou

Akcie pre hráča bez lopty:

- pohyb do jedného z ôsmich možných smerov z aktuálnej pozície
- pohyb do jedného z ôsmich možných smerov z domovskej pozície

- pohyb do domovskej pozície

Pri výbere konkrétnej akcie hráč zohľadňuje prioritné triedy jednotlivých akcií. Priority akcií sú logicky určené tak, aby ich výsledok bol čo najlepší. Poradie priorít je nasledovné:

- 1) strela na bránu
- 2) nahrávka spoluhráčovi dopredu (ak je viac možností, tak tomu, ktorý je bližšie k súperovej bráne)
- 3) dribling s loptou
- 4) nahrávka spoluhráčovi dozadu
- 5) podržanie lopty

2.6.4 Určovanie pozície hráčov

Hráči získavajú informácie o svojej pozícii na základe vizuálnej informácie o pevných referenčných bodoch. Táto informácia je zašumená. Na určenie vlastnej pozície sa obyčajne používa algoritmus zisťujúci priesečníky kružníc so stredmi v jednotlivých referenčných bodoch a polomerami danými „videnou“ vzdialenosťou od tohto bodu. Tím Brainstormers navrhol vlastný algoritmus určovania pozície hráča založený na pravdepodobnostnom modeli umiestnenia hráča. Hráč sa nachádza na množine pozícií ohodnotených váhami. Tím dosiahol s využitím tohto algoritmu až 60-percentné zvýšenie presnosti odhadu pozície. V štandardných situáciách (rohy, auty, rozohrávanie lopty) je však algoritmus s využitím kružníc rýchlejší.

2.7 UvA TRILEARN

2.7.1 Základné informácie

Tím UvA Trilearn vznikol v roku 2001 na pôde Amsterdamskej univerzity a dnes patrí k európskej a svetovej špičke. Hráč bol vyvíjaný úplne od začiatku a neobsahuje v sebe žiaden kód hráča iného tímu.

V roku 2001 sa autori zamerali najmä na nízku úroveň hráča, a snažili sa vylepšiť algoritmy použité inými špičkovými tímami (najmä FC Portugal). Hlavnou úlohou tímu bolo vylepšiť synchronizáciu hráčov a spresniť metódy určovania polohy a rýchlosti. Taktiež sa venovali aj vyššej úrovni myslenia hráča a to najmä strategickému uvažovaniu hráča a tvorbe kouča. Na RoboCup2001 Seattle sa tím umiestnil na výbornom štvrtom mieste.

V roku 2002 boli do kódu hráča pridané viaceré vylepšenia a to najmä pri určovaní polohy a rýchlostí objektov, kde bol použitý časticový filter. Taktiež bola vylepšená spolupráca hráčov, prioritno-dôveryhodný výber akcií. Na RoboCup2002 Fukuoka sa tím umiestnil na štvrtom mieste.

V roku 2003 sa pokračovalo vo vylepšovaní vlastností hráča z roku 2002. Do hráča pribudli vylepšené algoritmy na zachytávanie prihrávk, využívanie prihrávk a koordináciu akcií hráčov pomocou koordinačných grafov. Na RoboCup2003 Padova sa

tím stal absolútnym svetovým šampiónom. V ten istý rok sa stal aj víťazom American Open 2003 a German Open 2003.

V roku 2004 sa pokračovalo vo vylepšovaní vlastností hráča z roku 2003. Hlavné vylepšovanie sa sústredilo na koordináciu medzi jednotlivými hráčmi využívajúc koordinačné grafy a aplikovanie tohto modelu na predikáciu správania sa spoluhráča. Na RoboCup2004 Lisabon sa tím umiestnil na siedmom mieste.

V roku 2005 sa pokračovalo vo vylepšovaní vlastností hráča z roku 2004. Vývojový tím sa zamerl najmä na štúdium a vylepšovanie koordinácie medzi dvoma hráčmi používajúc pritom aproximačné alternatívy pre elimináciu premenných v koordinačných grafoch a aplikáciu tohto modelu na pre koordináciu hracieho tímu ako celku. Na RoboCup2005 Osaka sa tím umiestnil na desiatom mieste.

2.7.2 Analýza hráča

Architektúra hráča z pohľadu vrstiev je nasledovná:

- Spodná vrstva – jej úlohou je spracovanie vnímania a konania
- Skills layer –zabezpečuje základné vlastnosti hráča (dribbling, základné uvažovanie)
- Control Layer –výber najlepšej akcie spomedzi všetkých možných akcií hráča

Zachytávanie lopty a prihrávk

Hráč vníma svet pomocou svojich senzorov. Keďže informácie zo zrkového vnemu prichádzajú hráčovi v zašumenej podobe, tím sa rozhodol použiť **časticový filter**, ktorý čiastočne pomáha odstraňovať tento šum. Informácie o polohách a rýchlostiach jednotlivých objektov si hráči vymieňajú pomocou zvukových správ. Týmto spôsobom získava hráč presnejšiu informáciu o dianí na ihrisku. Na základe všetkých informácií potom hráč vyhodnotí situáciu na ihrisku a zmení svoj stav.

Na určenie miesta, v ktorom sa má hráč pokúsiť zachytiť loptu sa využíva algoritmus, ktorého súčasťou je časticový filter. Vo výpočte algoritmu je zahrnutá aj možnosť, že súperov hráč dobehne loptu skôr.

Podobne sa postupuje aj pri rozhodovaní pri prihrávkach, kde sa uvažujú také miesta v dráhe lopty, kde ju môže zachytiť spoluhráč, ale aj miesta, kde ju môže zachytiť súper.

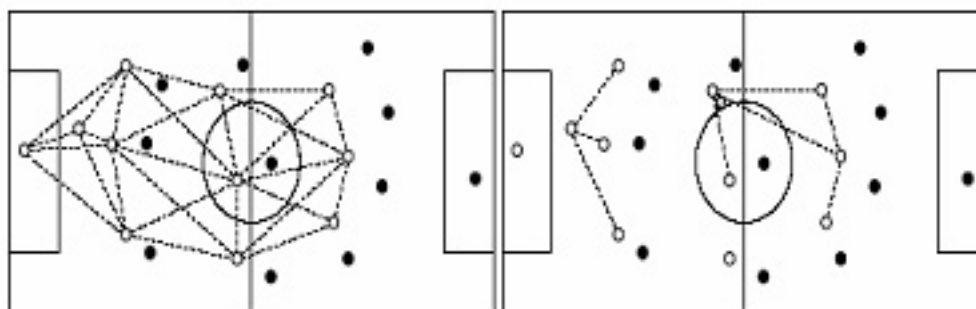
Ak má hráč schopnosť skórovať postupuje sa tak, že najskôr sa určí pravdepodobnosť, či strela skončí v bránke pri vystrelení z konkrétneho miesta, potom sa určí pravdepodobnosť, či brankár zachytí strelu z daného miesta na bránku.

Koordinácia hráčov na ihrisku

Hráč používa informáciu o dianí na ihrisku aj na predikciu správania sa spoluhráča a koordináciu akcií. Koordinácia medzi hráčmi je riešená pomocou **koordinačných grafov**. Reprezentácia koordinačných grafov je nasledovná:

- Uzol grafu – reprezentuje agenta (hráča)

- Hrana grafu – reprezentuje koordináciu medzi dvoma hráčmi (uzlami grafu), presnejšie povedané títo hráči musia skordinovať svoje akcie.



(a) Complete coordination Graph.

(b) Possible coordination graph after role distribution.

Obr. 2-4 Rozpad koordinačného grafu na skupinu podgrafov a nahradenie agentov rolami hráčov

Keďže však výpočtová zložitosť, ktorá určí optimálnu akciu pre jednotlivých agentov pri koordinácii rastie exponenciálne s počtom uzlov grafu, berie sa pri každom hráčovi do úvahy iba malá skupinka spoluhráčov okolo neho. Ďalšia úprava spočíva v nahradení spojitého stavového priestoru diskretným priestorom a s tým súvisiacou náhradou agentov rolami, tj. nebudú sa koordinovať agenti ale role hráčov. Samotná koordinácia prebieha pomocou posielania správ medzi hráčmi. Na elimináciu premenných v grafe bol v poslednej verzii kódu (rok 2005) implementovaný *Max-Plus algoritmus*, ktorý dokáže nájsť riešenie ďaleko rýchlejšie. Pri výbere akcie, ktorá sa má vykonať sú porovnané všetky akcie a ohodnotené pomocou miery dôveryhodnosti. Tá sa vypočíta z priority akcie v kombinácii a z pravdepodobnosti splnenia tejto akcie za daných podmienok (dôveryhodnosť akcie). Vyberie sa akcia s najväčšou mierou dôveryhodnosti. Silnou stránkou tohto prístupu je, že sa uvažujú všetky možnosti pre danú situáciu.

2.7.3 Zhrnutie

Tím UvA Trilearn patrí dnes medzi európsku a svetovú špičku. Na potlačenie šumu spôsobeného serverom používa časticový filter. Tento algoritmus sa javí ako veľmi efektívny a preto by bolo zaujímavé ho využiť aj v našom hráčovi. Pri koordinácii jednotlivých hráčov sa používajú tzv. koordinačné grafy, ktoré boli začlenené do kódu v roku 2003, kedy sa tím stal majstrom sveta. Na urýchlenie výpočtu nájdenia optima koordinačného grafu bol použitý Max-Plus algoritmus.

2.8 FC PORTUGAL 2001

Hráč je výsledkom spolupráce Univerzity Porto a Aveiro. Zamerali sa na zlepšovanie schopností hráča pomocou kouča, či už pred samotnou hrou alebo počas nej. Stratégia je definovaná v konfiguračnom súbore a môže ju počas hry meniť kouč.

2.8.1 Stratégia

Každá taktika obsahuje niekoľko globálnych parametrov a niekoľko formácií používaných v špecifických situáciách. Konfiguračný súbor obsahuje aj informácie o súperovom hráčovi, takže sa dá nastaviť určená stratégia. Stratégia je členená podľa situácie: základná stratégia, stratégia pre hru s loptou, stratégia pre získanie lopty.

Autori sa sústredili na vysokoúrovňové funkcie, nízkoúrovňové funkcie sa v podstate prebrali od pôvodného tímu (CMUnited99). Hráči kričia iba v tom prípade, ak uznajú, že informácia je pre tím dôležitá.

Pri útoku sa používa otvorená formácia, čím je možné vytvoriť viacero možností na strelbu. Naopak, pri obrane sa použije uzavretá stratégia, kedy sú hráči koncentrovaní okolo stredu ihriska.

2.8.2 Rozmiestňovanie

Hlavným vylepšením hráča je použitie SBSP (Situation Based Strategic Positioning). V prípade, ak hráč nepovažuje situáciu za kritickú, vráti sa na svoju strategickú pozíciu. Výhodou tohoto prístupu je, že hráči majú informáciu o pozícií svojich spoluhráčov aj v prípade, ak ich priamo nevidia. Ďalej je lepšie rozloženie hráčov na ihrisku, čím sa docieli zachovanie obranných pozícií a zlepšia možnosti na útok.

2.8.3 Výmena rolí

DPRE (Dynamic Positioning and Role Exchange) umožňuje výmenu rolí medzi jednotlivými hráčmi, ktorí sú homogénni. Táto informácia musí byť oznámená všetkým hráčom a uskutoční sa len v prípade, ak je to pre tím výhodné.

2.8.4 Pozorovací mechanizmus

Informácie zo senzorov sa vyhodnocujú podľa danej situácie, je im pridaná určitá váha, na základe ktorej sa potom hráč rozhoduje.

2.9 VÝBER HRÁČA

Po analyzovaní vybraných tímov sme sa rozhodli pokračovať v práci tímu FC Farmári. Hlavným dôvodom sú výborné výsledky z turnaja a schopností hráča. Napriek tomu si nechávame priestor pre možnú zmenu tohto výberu v závislosti od výsledkov dosiahnutých počas vývoja prototypu.

3 ŠPECIFIKÁCIA

Cieľom tejto časti je špecifikácia možných vylepšení existujúceho hráča, ktoré by mali prispieť k lepšej hre. Obsahuje funkčné požiadavky, ktorým sa budeme venovať.

3.1 NEURÓNOVÉ SIETE

Tím FC Farmári úspešne implementovali neurónovú sieť, ktorá dokázala nahradiť pôvodný reaktívny algoritmus pri určovaní, či sa má daná prihrávka uskutočniť alebo nie. Sieť sa inicializovala z jedného vstupného súboru, ktorý vznikol ako výstup programu HotParser. HotParser vytvára vstupný súbor na základe parsovania výstupov súborov vytvorených jednotlivými hráčmi pri prihrávkach počas tréningu. Vstup neurónovej siete má formát

$$d1 \quad d2 \quad d3 \quad y$$

(viď obr.), kde $d1$ je priama vzdialenosť od hráča, ktorému je určená prihrávka, $d2$ je priama vzdialenosť (v smere prihrávky) protihráča, $d3$ je kolmá vzdialenosť protihráča a y je vzor neurónovej siete, ktorý určuje či je daná prihrávka úspešná alebo neúspešná. Po jej vytvorení si sieť načíta každý hráč a na základe vstupného vektoru mu sieť určí pravdepodobnosť s akou bude prihrávka úspešná. Hráč vkladá na vstup siete pre danú polohu spoluhráča polohy všetkých protihráčov, ktorých vidí. Hráč realizuje prihrávku iba vtedy, ak sú všetky tieto pravdepodobnosti väčšie ako je medzná hodnota (pôvodne určená na 0,6). Na základe týchto informácií navrhujeme nasledujúce vylepšenia.

3.1.1 Použitie viacerých neurónových sietí

Keďže každá neurónová sieť používa pri svojej inicializácii rôzne náhodné vektory váh jednotlivých neurónov a aj po naučení s konečnou presnosťou sú tieto neurónové siete rôzne, budú aj pre rovnaké vstupy dávať rôzne pravdepodobnosti úspešnosti prihrávok. Taktiež je nutné si uvedomiť, že pri tréningu sa hrá proti vlastnému tímu a teda daná neurónová sieť bude len málo pripravená na hru proti inému súperovi, keďže správanie jeho hráčov pri zachytávaní prihrávok môže byť úplne odlišné. Preto by bolo vhodné použiť pri reálnom zápase viac neurónových sietí, z ktorých každá bude iná a hráč na základe presného algoritmu určí, ktorá sa bude používať.

3.1.2 Neurónové siete naučené na základe rôznych súperových formácií

Keďže nie je možné určiť, ako sa budú súperovi hráči správať pri našich prihrávkach je vhodné použiť aspoň klasifikáciu podľa aktuálnej súperovej formácie. Tá nám dáva odhad štýlu hry, ktorým sa riadi súper (napr. pri obrannom štýle hry má súper veľa obrancov, čo spôsobuje zachytávanie prihrávok súperom). Tím FC Farmári dokáže klasifikovať celkovo 20 formácií, ale použiť takýto počet neurónových sietí by bol veľmi nákladný a neefektívny. Preto sa obmedzíme len na rozdelenie podľa základných formácií, ktorých je

6. Vytvorenie vstupu pre HotParser bude spočívať v tom, že nami trénovaný tím postavíme proti súperovmu, ktorý bude mať hráčov v určenej hlavnej formácii (napr. 3-2-5). Kouč trénovaného tímu bude mať možnosť ľubovoľne meniť formácie a kouč súpera bude mať možnosť meniť iba subformácie danej hlavnej formácie. Týmto spôsobom sa dosiahne väčšia flexibilita pri vytváraní tréningového vzoru pre neurónovú sieť. Takto získame celkovo 6 rôznych neurónových sietí, z ktorých každá bude prispôbená jednej z hlavných formácií a bude mať v sebe zakomponované aj prípadné subformácie.

Ako alternatívu možno uvažovať vytvorenie konkrétnych neurónových sietí pre danú pozíciu hráča v tíme (obranca, stredopoliar, útočník). Hráč v konkrétnej pozícii určenej koučom by si mohol vyberať iba neurónové siete z určitej množiny. Táto alternatíva má výhodu oproti pôvodnej hlavne v tom, že uvažuje nad samotnou pozíciou hráča v tíme. Napríklad keď si uvedomíme, že útočnickove prihrávky budú mať v prehustenej obrane menšiu pravdepodobnosť ako prihrávky obrancu.

Výber konkrétnej neurónovej siete, ktorá bude určovať pravdepodobnosť prihrávky je určená

- koučom – na základe analýzy súperových formácií
- hráčom – na základe učenia sa pomocou odmeny a trestu.

3.2 PRIDANIE AKCIÍ

Akciu definujeme ako postupnosť krokov, ktoré majú vykonať viacerí hráči. Základom pre riadenie je zvuková komunikácia. Akciu definuje hráč, ktorý je najbližšie k lopte. Ak hráč rozozná situáciu, v ktorej pozná akciu, tak v tomto prípade oznámi začatie akcie a hráčov, ktorí sa jej zúčastnia. Vykonanie akcie dostane najvyššiu prioritu. V prípade, ak hráč nerozoznal akciu, pokračuje sa podľa rozhodovania pôvodného hráča.

Na rozoznanie danej situácie sa použije porovnávanie vzorov. Identifikácia sa bude skladať z viacerých krokov:

- rozpoznanie rozmiestnenia spoluhráčov pomocou vzorov
- ak sa našiel vzor, je k nemu priradená maska, ktorá určuje oblasť, v ktorej sa nesmú nachádzať protihráči
- k maske sú pridané ďalšie podmienky, ako napríklad pozícia na ihrisku, pozícia brankára
- ak dostaneme pozitívny výsledok, môžeme vyhlásiť akciu.
- jednotlivé kroky, ktoré musí vykonať daný hráč sú definované pre každú akciu.

Po vyhodnotení akcie sa zvyčajne vykoná úloha s loptou a až potom sa oznámi akcia. Po prijatí lopty ďalším hráčom tento opäť vyhodnocuje situáciu pomocou vzorov. Ak sa k lopte nedostal, akciu zruší.

Štandardné akcie, ktoré by sa takýmto spôsobom dali ošetriť je prihranie hráčovi pomocou asistencie iného voľného hráča, ktorý sa nachádza viac vzadu; určenie voľného hráča apod.

Na definovanie by bolo vhodné vytvoriť editor a ďalej analyzátor, ktorý by vytváral masku.

Maska predstavuje takú polohu protihráčov, z ktorej môžu prekaziť prihrávku s určitou pravdepodobnosťou v prípade, ak by vedeli presnú dráhu lopty.

3.3 LIBERO

Libero je podľa [Forma2005] „futbalový obranca voľne zasahujúci do hry po celom ihrisku“. Spravidla býva ako libero vybraný starší skúsený hráč, ktorý nemá pevne určené miesto v rámci formácie, ale sám si vyberá svoju pozíciu na ihrisku v závislosti od vývoja hry a aktuálnej potreby.

Chceli by sme sa pokúsiť nasimulovať správanie takéhoto hráča. Jeden hráč na pozícii stredopoliara sa v prípade hry pred vlastnou bránou, keď loptu má súper, presunie na pozíciu obrancu, aby prehustil priestor pred vlastnou bránou a vypomohol obrane. Zvolený stredopoliar musí analyzovať situáciu. V prípade jej pozitívneho vyhodnotenia informuje (zvukovou komunikáciou) ostatných spoluhráčov v strede poľa, aby pozmenili svoje rozostavenie tak, aby zaplnili voľný priestor, ktorý po ňom zostane. Tiež musí informovať obrancov, aby zmenili svoje rozostavenie s ohľadom na nového „pomocníka v obrane“. Po získaní lopty a presune ťažiska hry ďalej od vlastnej brány libero odvolá akciu, presunie sa naspäť do stredovej formácie a pošle zvukový signál ostatným spoluhráčom, aby zaujali svoje pôvodné miesta.

Hlavnou myšlienkou navrhovanej stratégie je predpoklad, že hráč je potrebný skôr v blízkosti lopty než ďalej v strede ihriska.

Podobné libero sa pokúsime vytvoriť z niektorého stredopoliara aj v útočnej fáze hry. Táto taktika môže byť úspešná najmä vtedy, ak súperov tím dynamicky mení svoju formáciu podľa formácie druhého (teda nášho) tímu. V tomto prípade, ak je počet súperových obrancov určený podľa počtu našich útočníkov, tak v prípade rýchleho a nečakaného zapojenia ďalšieho hráča na hrot útoku môže dôjsť k prečísleniu a tým k zvýšeniu šancí na strelenie gólu.

Pri využívaní „útočného libera“ je dôležité nezabudnúť na zadné vrátka, aby nedošlo k prečísleniu na opačnej strane. Predpokladáme však, že výhoda získaná zapojením väčšieho počtu hráčov do útoku bude väčšia než nevýhoda spôsobená menším počtom hráčov vzadu pri strate lopty. Keď tím príde o loptu pred súperovou bránou, je ešte dost času, aby sa libero vrátil do svojej pozície než sa lopta dostane do časti ihriska, odkiaľ by mohla vzniknúť akcia na ohrozenie našej brány.

Ako útočné a obranné libero nebude zvolený ten istý hráč zo stredu poľa, aby sa príliš nevyčerpal neustálym prebiehaním z jednej strany ihriska na druhú. Na post libera plánujeme využiť heterotyp hráča, ktorému sa rýchlejšie dopĺňa výdrž (stamina).

3.4 ŠTATISTICKÁ ANALÝZA ROZLOŽENIA HRÁČOV SÚPERA

Naším ďalším návrhom je, aby kouč vykonával štatistickú analýzu rozloženia súperových hráčov na ihrisku. S využitím tejto štatistiky môžu byť naši útočiaci hráči nasmerovaní do priestoru, kde sa súperovi obrancovia vyskytujú s malou pravdepodobnosťou (napr. preto, že sa striktne zdržujú v okolí svojich domovských pozícií vo formácii alebo preto, lebo nesprávne pokrývajú plochu pred vlastnou bránou). Kouč bude niekoľkokrát za zápas informovať hráčov o vhodných prázdnych priestoroch na smerovanie prihrávky. Implementáciou tejto stratégie získa hráč, ktorý dostane loptu vo voľnom priestore, viac času na rozmyslenie a výber ďalšej akcie (streľba alebo vhodná nahrávka).

V prototypy sa pokúsime overiť, nakoľko je toto vylepšenie zakomponovateľné do útočnej fázy hry vylepšovaného hráča.

Táto stratégia môže byť skombinovaná s vyššie uvedenou stratégiou „útočného libera“. Libero pribiehajúce na post útočníka si môže vybrať vhodné prázdne miesto v súperovej obrane a môže si zvukovým povelením vypýtať od spoluhráča s loptou prihrávku na danú pozíciu, pričom tam sám beží na 100%.

4 HRUBÝ NÁVRH

V tejto kapitole sa zameriavame na hrubý opis algoritmov a princípov, ktoré chceme použiť pri implementácii prototypu a výsledného produktu. Zameriavame sa hlavne na vylepšenie neurónových sietí, rozpoznávanie pozícií hráčov pre vyvolanie akcie a použitie libera.

4.1 NEURÓNOVÉ SIETE

Táto časť návrhu sa zaoberá samotným algoritmom pri určovaní konkrétnej neurónovej siete hráčom. Boli navrhnuté dve odlišné koncepcie a to výber na základe informácií od kouča a výber na základe rozhodnutia sa hráča.

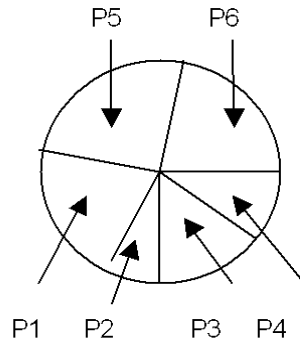
4.1.1 Výber na základe informácií od Kouča

Kouč v tíme FC Farmári na základe informácií o rozložení súperových hráčov na ihrisku klasifikuje súperove formácie a následne po každých 300 cykloch upravuje formáciu vlastného tímu. Je vhodné posielat' aj informáciu o tom, aká neurónová sieť sa má použiť. Tento spôsob je veľmi jednoduchý a pohodlný, ale na druhej strane málo flexibilný. Súper môže počas zápasu úplne zmeniť formáciu a tým sa určená neurónová sieť stane nevyhovujúcou.

4.1.2 Výber na základe rozhodnutia sa hráča

Spôsob výberu neurónovej siete na základe rozhodnutia sa hráča je veľmi efektívny najmä kvôli flexibilitě, ktorú jednotlivým typom hráčov (obranca, stredopoliar, útočník) poskytuje. Taktiež je vhodný aj pre rýchle adaptovanie sa na novú stratégiu hry súpera. Navrhujeme preto vytvoriť dva rôzne algoritmy, ktoré by vybrali neurónovú sieť na základe úspešnosti danej prihrávky.

Algoritmus s použitím pseudonáhodného rozdelenia



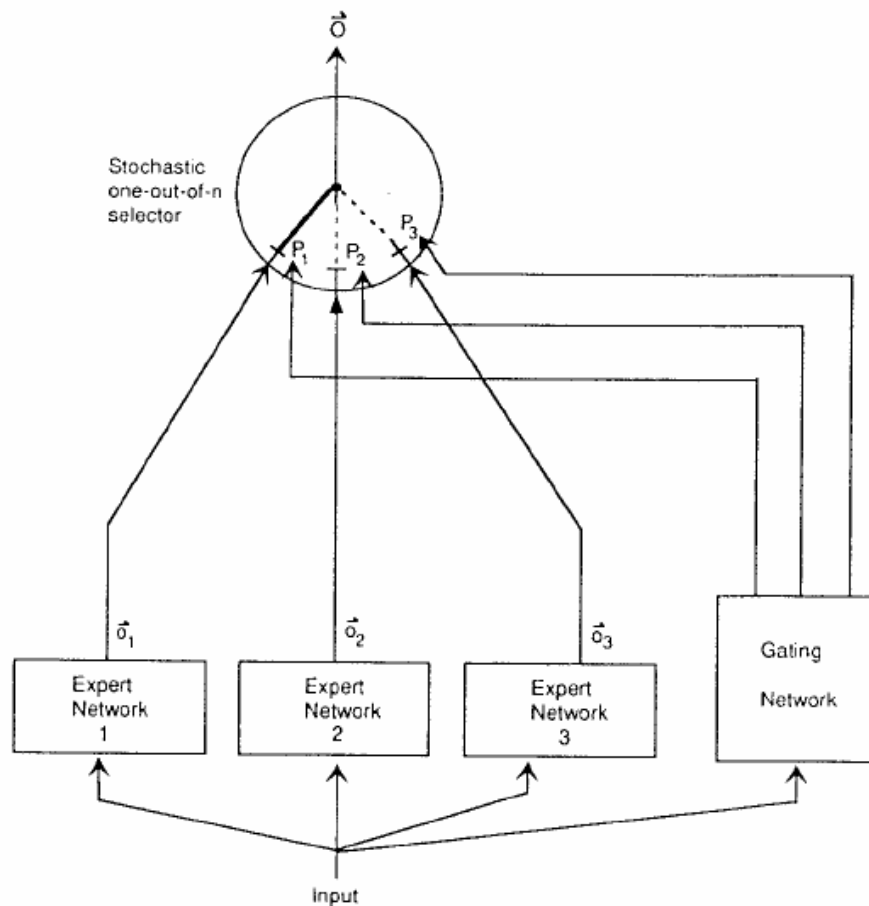
Obr. 4-1 Zobrazenie pravdepodobností výberu jednotlivých neurónových sietí

Tento algoritmus by fungoval na základe učenia sa podľa toho, či bola daná prihrávka úspešná alebo nie. Je založený na pseudonáhodnom výbere. Jedná sa o výber pomocou rulety ako pri genetickom algoritme. Každý neurónovej sieti je pridelený určitý interval pravdepodobnosti. Výber sa deje pomocou náhodného čísla z intervalu $<0,1>$, ktoré určí neurónovú sieť na výpočet pravdepodobnosti.

Hráč si vyberie neurónovú sieť a tá mu určí pravdepodobnosť úspešnosti prihrávky. Ak je pravdepodobnosť vyššia ako je prahová hodnota, hráč ju zrealizuje a zapamätá si to. Potom zisťuje či bola daná prihrávka úspešná alebo nie. Ak bola úspešná, zvýši sa pravdepodobnosť výberu danej neurónovej siete na úkor iných sietí. Ak nie, pravdepodobnosť sa zníži. Týmto postupom je zaručené, že siete, ktoré budú určovať pravdepodobnosť prihrávky veľmi dobre budú určené na posudzovanie častejšie a naopak. Ostáva len určiť hranice do akej miery môžu dané intervaly pravdepodobnosti rásť a klesať, najmä z dôvodu, aby sa pri zmene súperovej formácie dokázali intervaly pravdepodobnosti rýchlo upraviť. Navrhujeme, aby na začiatku mali všetky neurónové siete rovnakú pravdepodobnosť výberu. Následnou evolúciou systému by jednotlivé intervaly lineárne upravovali tak, aby najnižšia pravdepodobnosť výberu jednej siete bola 4% a najvyššia 80%. Tieto hodnoty sú však iba predbežné a môžu sa počas testovania prototypu hráča zmeniť. Tento spôsob výberu hráčom je možné ľahko kombinovať aj s výberom neurónovej siete na základe informácií od kouča, kedy by sa po prijatí informácie hráčom skokovo upravili intervaly pravdepodobnosti a ďalej by sa pokračovalo v pôvodnom algoritme.

Výber neurónovej siete pomocou algoritmu adaptívnej kombinácie lokálnych expertov

Algoritmus je alternatívou k algoritmu pseudonáhodného rozdelenia. Hlavnou myšlienkou je, že nad neurónovými sieťami bude iná neurónová sieť (tzv. rozhodovacia), ktorá na základe vstupov určí, aká sieť bude klasifikovať úspešnosť prihrávky.

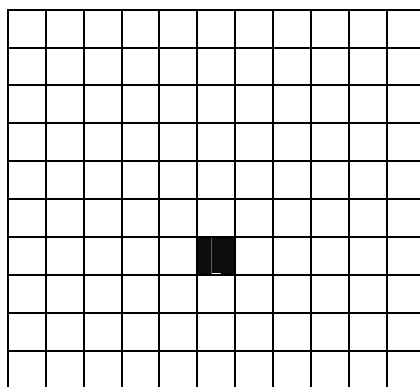


Obr. 4-2 Výber výstupu konkrétnej neurónovej siete na základe určenia pravdepodobnosti rozhodovacou sieťou

Vstup rozhodovacej siete môže byť rovnaký, ako u sietí určujúcich pravdepodobnosti prihrávkov sietí. Výstupom siete je normalizovaný vektor, ktorého jednotlivé zložky určujú pravdepodobnosť použitia konkrétnej neurónovej siete. Ide teda o istý druh stochastického prepínača, ktorý určuje pravdepodobnosť výberu výstupu určitej neurónovej siete. Rozhodovacia sieť sa bude učiť algoritmom BackPropagation na základe úspechu, resp. neúspechu realizovanej prihrávky kvalifikovanej konkrétnou vybranou neurónovou sieťou. Tento algoritmus je vhodnejší ako samotné upravovanie váh jednotlivých sietí, lebo v najhoršom prípade nebude celkom správne klasifikovať vybranú sieť. Pri adaptácii váh samotných sietí by sa mohlo stať, že sa už naučené siete ďalším učením pokazia. Vzhľadom na to, že algoritmus je pomerne nový, je ťažké odhadnúť nakoľko bude úspešný. V prípade jeho implementácie budeme musieť zvážiť rozšírenie vstupného vektora rozhodovacej siete a vhodné nastavenie počiatkových váh, aby žiadna z vybraných sietí nebola na začiatku zápasu znevýhodnená.

4.2 POROVNÁVANIE POMOCOU VZOROU

Hráč si vyčlení časť ihriska takým spôsobom, že sa bude nachádzať v jednej tretine (Obr. 4-3). Táto časť sa rozdelí na mriežku s rovnakými vzdialenosti. Podľa svojich informácií o polohe spoluhráčov určí polohu v mriežke. Označíme ju ako prvý druh.



Obr. 4-3 Umiestnenie hráča s loptou v mriežke

Takto vyplnenú mriežku porovná s uloženými vzormi rovnakých rozmerov. Všetky vyplnené miesta na mriežke vo vzore sa musia prekryť s hráčom vyplnenou mriežkou. Ostatné pozície nás nezaujímajú. Takto získame množinu vzorov, ktoré sa budú ďalej rozpracovávať. Pre každý jeden vzor existuje aspoň jedna maska, ktorá môže mať iné parametre ako prvá mriežka. Úlohou tejto mriežky je určiť pozície súperových hráčov, označíme ju ako mriežka druhého druhu. Aplikovaním mriežky zistíme, či sa v určitom priestore nenachádzajú súper. Takto dostávame kombinácie mriežok prvého a druhého druhu. Táto kombinácia určuje východiskovú situáciu pre určitú akciu.

Východisková situácia musí spĺňať ďalšie podmienky, ako je napríklad pozícia v ofsajde, pozícia brány. Po vyhodnotení máme situáciu, z ktorej môžeme začať akciu. Avšak k dispozícií môže byť viacero východiskových situácií.

Ďalším krokom je ohodnotenie situácií pomocou váh. K dispozícií máme hodnotu, kam by sa mala lopta dostať po ukončení akcie. Keďže väčšina by mala končiť v bráne, ako ďalšie kritérium zoberieme počet hráčov zapojených do akcie.

Už máme zvolenú akciu, teraz je potrebné ju vykonať.

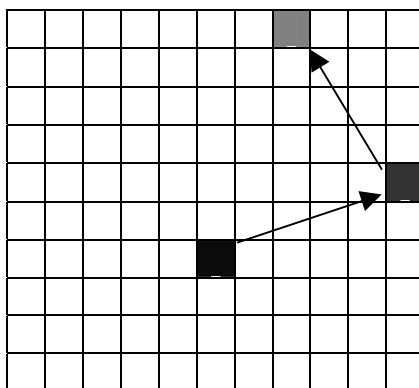
4.3 AKCIA

Ide o postupnosť krokov, ktoré sú vykonávané na základe preddefinovaných údajov. Tieto údaje budú vytvorené pred samotnou hrou a budeme ich definovať ručne (špecifické akcie, ktoré sme schopní zdefinovať) alebo generovať automaticky na základe analýzy herného postavenia.

Akciu spúšťa hráč, ktorý je najbližšie k lopte, výkrikom, ktorý ju definuje. Ostatní hráči sa môžu sústrediť na hráča s loptou, aby ho lepšie počuli. Okrem identifikátora akcie určuje nasledujúceho hráča, ktorý pokračuje v akcii.

Pri získaní lopty druhým hráčom tento testuje, či môže ďalej pokračovať v akcii, prípadne zmeniť akciu. Postupuje rovnakým spôsobom ako prvý hráč. V podstate nejde o pôvodnú akciu, ale nejaký jej fragment.. Ak hráč loptu nezískal a má ju súper alebo iný spoluhráč, potom odvoláva akciu. Aby sa predišlo vzájomnému protipôsobeniu akcií, kedy by si dvaja hráči navzájom len kopali loptu dookola, definujeme čas, počas ktorého sa nemôže akcia opakovať.

Postupnosti krokov v akcií sú zadané vo vstupnom súbore spolu s mriežkami. Akcie by mali byť hierarchické, mali by na seba nadväzovať (Obr. 4-4). Teda ak je definovaná akcia pre troch hráčov, po prijatí lopty druhým musí byť tento schopný identifikovať príslušnú akciu, ale už len pre dvoch hráčov. Akcie nebudú komplikované, väčšinou vzniknú zložením primitívnych akcií.



Obr. 4-4 Hierarchické akcie

Ak nie je možné žiadnu akciu vykonať, postupuje sa podľa upraveného hráča pôvodného tímu.

4.4 GENEROVANIE VZOROV A AKCIÍ

Okrem ručného zadefinovania vzorov a vytvorenia akcií môžeme vytvoriť nástroj, ktorý bude pre každú pozíciu spoluhráča vytvárať mriežku druhého druhu, ktorú dostane ako množinu bodov, z ktorých nie je možné pre protivníka chytiť loptu ani za predpokladu, že súper pozná presnú dráhu lopty.

Ďalšie východiskové situácie môžeme vytvoriť tak, že pre každú cieľovú pozíciu lopty zistíme políčka, kde sa nesmie nachádzať súper, ale môže sa nachádzať spoluhráč, ktorý je schopný prihrávku chytiť. Takto je možné vytvoriť akcie aj pre bránených hráčov.

4.5 LIBERO – ROZPOZNANIE SITUÁCIE

V tejto kapitole navrhujeme algoritmus na rozpoznanie situácie, v ktorej sa má libero presunúť zo stredovej formácie do obrany, resp. útoku, a informovať o novej hernej situácii spoluhráčov.

4.5.1 Presun obranného libera do obrany

Obranný libero sa presunie do obrannej formácie za súčasného splnenia nasledujúcich podmienok:

- loptu ovláda súper
- obranný libero je vzdialený od lopty viac ako vzdialenosť $L1$
- lopta sa nachádza vo vzdialenosti najviac $B1$ od brány nášho tímu

- počet súperových hráčov, ktorých libero vidí v blízkosti našej brány (t.j. vo vzdialenosti max. S1 od brány) alebo videl v priebehu posledných dvoch cyklov, je aspoň N1
- obranný libero nie je vyčerpaný na viac ako P1%

Konštanty L1, B1, S1, N1 a P1 určíme počas tréovania hráča. Ako počiatočné hodnoty budeme uvažovať:

L1 = 15 metrov

B1 = $\frac{1}{4}$ dĺžky ihriska

S1 = $\frac{1}{4}$ dĺžky ihriska

N1 = 4

P1 = 40

4.5.2 Presun obranného libera naspäť do stredovej formácie

Obranný libero sa presunie naspäť do stredovej formácie za súčasného splnenia nasledujúcich podmienok:

- loptu ovláda spoluhráč, ale nie sám obranný libero
- lopta sa nachádza vo vzdialenosti aspoň B2 od brány nášho tímu
- obranný libero nie je vyčerpaný na viac ako P2%

Konštanty B2 a P2 určíme počas tréovania hráča. Ako počiatočné hodnoty budeme uvažovať:

B2 = $\frac{1}{2}$ dĺžky ihriska

P2 = 30

4.5.3 Presun útočného libera do útoku

Útočný libero sa presunie do útočnej formácie za súčasného splnenia nasledujúcich podmienok:

- loptu ovláda spoluhráč
- lopta sa nachádza vo vzdialenosti najviac B3 od brány súperovho tímu
- útočný libero nie je vyčerpaný na viac ako P3%

Konštanty B3 a P3 určíme počas tréovania hráča. Ako počiatočné hodnoty budeme uvažovať:

B3 = $\frac{1}{4}$ dĺžky ihriska

P3 = 40

4.5.4 Presun útočného libera naspäť do stredovej formácie

Útočný libero sa presunie naspäť do stredovej formácie za súčasného splnenia nasledujúcich podmienok:

- loptu ovláda súper

- lopta sa nachádza vo vzdialenosti aspoň B4 od brány súperovho tímu
- útočný libero nie je vyčerpaný na viac ako P4%

Konštanty B4 a P4 určíme počas tréningu hráča. Ako počiatočné hodnoty budeme uvažovať:

B4 = 1/3 dĺžky ihriska

P4 = 30

4.6 ŠTATISTICKÁ ANALÝZA ROZLOŽENIA HRÁČOV SÚPERA

Štatistickú analýzu rozloženia hráčov súpera na ihrisku bude vykonávať kouč. Zameria sa na priestor pred súperovou bránou. Získané štatistické údaje vyhodnotí záplavovým alebo výškovým algoritmom. Oba tieto algoritmy sú popísané v [Squirrel2005]. Výsledkom analýzy bude niekoľko odporúčaných pozícií pre našich útočníkov, ktoré sú málo obsadzované súperovými obrancami a pritom ich poloha je dôležitá.

Kouč sprostredkuje získané informácie hráčom (počas polčasovej prestávky alebo prerušenia hry alebo dva krát za polčas). Pri definovaní akcií naši útočníci hráči zohľadnia informácie o odporúčaných výhodných pozíciách. Útočný libero pri presune do útočnej formácie v prípade, že sa nachádza v blízkosti takejto pozície, zakričí svojmu spoluhráčovi s loptou správu *Prihraj* spolu s jej súradnicami a sám beží na túto pozíciu na 100%. Toto je len hrubý návrh novej stratégie. Presnejšie algoritmy identifikácie vhodnosti vyhlásenia útočnej akcie určíme počas tréningu.

4.7 PRIHRÁVKY OBSADENÝM HRÁČOM

Útočník s loptou sa nachádza v obrannom pásme súpera a pred sebou nevidí žiadnych voľných hráčov. Dostáva sa do situácie, keď musí loptu okamžite odkopnúť, inak o ňu pravdepodobne príde. Na krídle vidí spoluhráča, ktorý je ale obsadený súperovým hráčom. Zakričí správu „Bež!!“ a obsadený hráč sa rozbehne maximálnou rýchlosťou dopredu, a zároveň hráč s loptou ju odkopne tak, aby ju spoluhráč mal šancu zachytiť a spracovať. Po odkopnutí lopty sa hráč rozbehne dopredu, aby mu mohol spoluhráč v prípade potreby nahráť (napríklad aj na jeden dotyk) a celý cyklus sa opakuje.

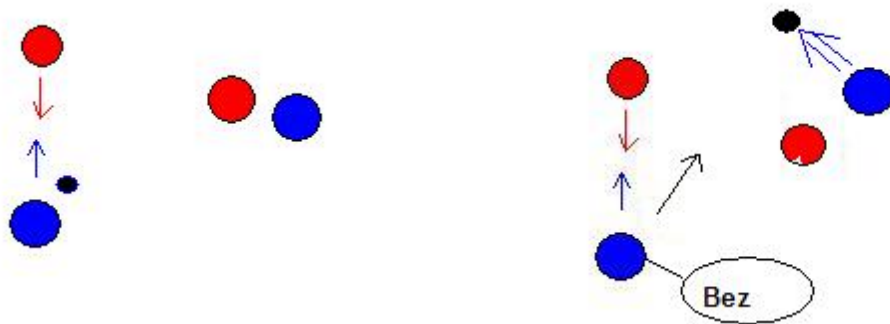
Situácia je zachytená na Obr. 4-5.

Opis algoritmu správania sa hráča:

1. Hráč strieľa na súperovu bránu, ak je možné strelit' gól
2. Hráč nahráva spoluhráčovi, ktorý je vo výhodnej útočnej pozícii
3. Hráč vedie loptu rýchlym driblingom
4. Hráč vedie loptu bezpečným driblingom
5. **Hráč prihráva obsadenému hráčovi (po jeho následnom uvoľnení)**
6. Hráč nahráva ľubovoľnému spoluhráčovi

Opis algoritmu správania sa hráča vo vyznačenej situácii:

1. Hráč skontroluje vysunutých obsadených hráčov
2. Vyberie si adresáta nahrávky
3. Pošle mu správu „Bež!!“ spolu s pozíciou, na ktorú má bežať
4. Hráč posiela loptu na zvolené miesto a **zároveň** sa adresát prihrávky **maximálnou rýchlosťou** dopraví na miesto určenia, čo by malo viesť k jeho uvoľneniu
5. Adresát prihrávky sa pokúša zachytiť loptu ,zatiaľ čo sa hráč, ktorý loptu odkopol snaží prejsť ďalej do útoku na takú pozíciu, aby mohol prijať prípadnú nahrávku



Obr. 4-5 Prihrávka obsadenému hráčovi

5 PROTOTYP

Obsahom tejto kapitoly sú výsledkom práce na prototypu, ktorého hlavnou úlohou bolo overenie možnosti implementácie navrhovaných zlepšení hráča tímu FC Farmári.

5.1 POUŽITIE NEURÓNOVÝCH SIETÍ PRI URČOVANÍ PRAVDEPODOBNOTÍ PRIHRÁVOK

5.1.1 Súčasný stav

Pri podrobnom skúmaní zdrojového kódu sme prišli k nasledujúcim skutočnostiam. Neurónová sieť, ktorá bola vytvorená tímom FC Farmári je inštanciou triedy *FCNetwork*. Meno premennej, ktorá reprezentuje sieť je *m_PassNet* a je atribútom triedy *PlayerTactics*. Táto sieť je dvojvrstvová so 45 skrytými neurónmi a s jedným výstupným neurónom. Po vytvorení siete sa načítajú váhy na jednotlivých neurónoch zo súboru *passvahy.txt* pomocou funkcie *LoadFrom*. Neurónová sieť bude používaná iba ak sa hráč spustí s argumentom *-pt*. To sa následne namapuje do premennej *PassTesting*. Rozhodovanie komu hráč prihrá sa určuje vo funkcii *GetPassAbleFriend*, kde sa na základe premennej *PassTesting* určí, či bude použitá neurónová sieť alebo reaktívny prístup. V prípade, že má byť použitá neurónová sieť vráti funkcia *GetPassAbleFriend* hodnotu funkcie *FCGetPassAbleFriend*, v ktorej sa vykoná samotné vyhodnotenie pomocou neurónovej siete. Vyhodnotenie sa uskutočňuje pomocou nasledujúceho algoritmu:

```
vykonaj pre všetkých spoluhráčov
  určí vzdialenosť spoluhráča od súperovej brány
  ak je vzdialenosť menšia ako moja vzdialenosť
    ak je vzdialenosť spoluhráča odomňa menšia ako 30m a väčšia ako 5m
      vykonaj pre všetkých protihráčov
      ak protihráč patrí do oblasti (FCBelongsToArea==true)
        určí potrebné vzdialenosti
        vykonaj výpočet neurónovou sieťou a určí pravdepodobnosť
        vynásob túto pravdepodobnosť s predchádzajúcimi pre daného
          spoluhráča
      ak protihráč nepatrí do oblasti
        vynásob 1 s predchádzajúcimi pravdepodobnosťami pre daného
          spoluhráča
    ak je aktuálna kumulovaná pravdepodobnosť väčšia ako najlepšia
      aktuálna sa stane najlepšia
  ak bolo čo vyhodnocovať
    vráť číslo hráča s najvyššou kumulovanou pravdepodobnosťou
  inak vráť -1
```

Ako je vidieť z opisu algoritmu, tak sa v ňom nikde nenachádza žiadna hraničná hodnota, ktorá by určovala minimálnu hranicu, kedy sa prihrávku oplatí realizovať. Toto je rozpor s pôvodnou dokumentáciou FC Farmári, kde takáto hraničná hodnota bola. Často sa potom stáva, že sa realizujú prihrávky aj s veľmi malou pravdepodobnosťou rádovo 0.001, čo

prakticky nemá zmysel. Pri podrobnejšej analýze kódu sme zistili, že pokiaľ funkcia *GetPassAbleFriend* vráti hodnotu -1 (čo značí, že neurónová sieť ani nemala čo vyhodnocovať, pretože súperovi hráči stáli „priamo“ v prihrávke alebo všetci spoluhráči boli ďalej od brány ako prihrávajúci hráč) vyberie sa vo funkcii *BehaveWithBall* najbližší spoluhráč, čo opäť nie je podľa nášho názoru správne, pretože je to príčina častých strácaní lôpt počas prihrávok. Z tohto dôvodu je nutné navrhnúť novú štruktúru funkcie *BehaveWithBall*.

5.1.2 Opis implementácie algoritmu na určovanie prihrávok

Na základe zistených skutočností sme vykonali nasledujúce úpravy funkcie *FCGetPassAbleFriend*. Zakomponovali sme do algoritmu hraničnú hodnotu (nazvanú *FM_stop* a je rovná hodnote 0.6), ktorá určuje hranicu, kedy má vôbec nejaký zmysel pokračovať v ohodnocovaní prihrávky pre konkrétneho spoluhráča. Ďalej sme zmenili výpočet kumulovanej pravdepodobnosti úspešnosti prihrávky, ktorá sa predtým počítala ako súčin ohodnotení neurónovou sieťou pre všetkých protihráčov. Pôvodný prístup nebol celkom v poriadku, pretože ak bola pravdepodobnosť prihrávky pre konkrétneho spoluhráča napr. 0,8 a to pre 4 protihráčov, potom kumulovaná pravdepodobnosť je $0,8^4 = 0,4096$. Toto nie je logicky správne, pretože táto hodnota je menšia ako pravdepodobnosť prihrávky pre iného spoluhráča, ktorého ohrozuje jediný protihráč na 0,5. Preto sme vytvorili nový systém výpočtu kumulovanej pravdepodobnosti, ktorá sa vypočíta ako najmenšia pravdepodobnosť ohodnotení neurónovou sieťou z pomedzi všetkých ohodnotení pre každého protihráča. Zavedením hraničnej hodnoty *FM_stop* sme taktiež zoptimalizovali aj výpočet, pretože pokiaľ nejaký protihráč ohrozuje prihrávku na viac ako je hodnota *FM_stop*, potom nemá zmysel vykonávať ďalšie ohodnocovanie pre daného spoluhráča a cyklus je nutné ukončiť. Ďalšie vylepšenie spočíva vo vyhľadávaní najbližšieho spoluhráča vzhľadom na súperovu bránu spomedzi tých, ktorý majú maximálne ohodnotenie. Prípád rovnakých ohodnotení pravdepodobností prihrávok nastáva iba vtedy ak sú nejaký dvaja spoluhráči prakticky voľný (t.j. funkcia *FCBelongsToArea* vráti false). Implementovaný algoritmus možno prepísať do štruktúrovaného textu takto:

```
vykonaj pre všetkých spoluhráčov
    urči vzdialenosť spoluhráča od súperovej brány
    ak je vzdialenosť menšia ako moja vzdialenosť
        ak je vzdialenosť spoluhráča odomňa menšia ako 30m a väčšia ako 5m
            vykonaj pre všetkých protihráčov
                ak protihráč patrí do oblasti (FCBelongsToArea==true)
                    urči potrebné vzdialenosti
                    vykonaj výpočet neurónovou sieťou a urči pravdepodobnosť
                    ak je táto pravdepodobnosť menšia ako najmenšia
                        kumulovaná pravdepodobnosť = nová pravdepodobnosť
                        ak je kumulovaná pravdepodobnosť menšia ako FM_stop
                            preruš cyklus a pokračuj ďalším spoluhráčom
                    ak protihráč nepatrí do oblasti
                        ak default pravdepodobnosť je väčšia ako 1
                            prirad' do kumulovanej pravdepodobnosti hodnotu 1
                ulož kumulovanú pravdepodobnosť
            najdi najvyššiu pravdepodobnosť, ktorá je priradená hráčovi s najmenšou
                vzdialenosťou od súperovej brány
        ak najlepšia kumulovaná pravdepodobnosť je väčšia ako FM_stop
            vráť číslo prislúchajúceho spoluhráča
```

5.1.3 Opis implementovanej neurónovej siete

Na základe zistených skutočností sme rozhodli vymeniť pôvodnú neurónovú sieť tímu FC Farmári a to najmä z nasledujúcich dôvodov:

- Pomerne ťažký spôsob tréningu siete, keďže v pôvodnom kóde N.S. sa nenachádzali žiadne podporné funkcie, ktoré by nám tréningu umožňovali
- Pochybná kvalita implementácie N.S., pretože sa veľmi často strácala lopta pri prihrávkach.
- Pomerne veľký počet skrytých neurónov v sieti (45), ktorý podstatne zvyšuje výpočtovú náročnosť pri ohodnocovaní prihrávk.

Samotná sieť je reprezentovaná inštanciou *FM_net* triedy *FM_Network*. Pointer *FM_net* sa nachádza v triede *PlayerTactics*. Pri spustení hráča sa v konštruktore *PlayerTactics* vytvorí objekt, pomocou konštruktora triedy *FM_Network*, ktorý má nasledujúci formát

FM_Network(pocet1,pocet2,pocet3,pocet4,"menosuboru")

Kde

- *pocet1* – počet skrytých neurónov, táto hodnota je v kóde nastavená na 20,
- *pocet2* – počet výstupných neurónov, táto hodnota je kóde nastavená na 1,
- *pocet3* – veľkosť vstupu pre vstupné neuróny, táto hodnota je kóde nastavená na 48 (3x16),
- *pocet4* – veľkosť výstupu pre skrytú vrstvu, táto hodnota je kóde nastavená na 1,
- *menosuboru* – je meno súboru, kde sú uložené váhy neurónovej siete, táto hodnota je kóde nastavená na reťazec *vahy.txt*.

Súbor *vahy.txt* sa vytvára pomocou programu **BackPropagation**, ktorý bude bližšie popísaný v používateľskej príručke.

Ako je vidieť vstupom neurónovej siete je „binárny“ reťazec dĺžky 48 a vytvára sa prekonvertovaním troch reálnych (pre každé číslo dĺžka 16) vzdialeností podľa formátu uvedenej v dokumentácii. Pri prototypovaní N.S. sme sa pokúšali dĺžku tohto reťazca zmenšiť. V prípade testovania reálnych vstupných hodnôt sa sieť nedokázala naučiť tréningovú množinu a počet zle klasifikovaných vzorov bol takmer polovica zo všetkých vzorov. V prípade testovania binárnych vstupov s dĺžkou 3x8 nebola dosiahnutá požadovaná presnosť, pretože kumulovaná chyba bola približne 16 a to aj pri pomerne veľkom počte skrytých neurónov. V prípade testovania binárnych vstupov s dĺžkou 3x12 sa síce dosiahnutá presnosť zväčšila ale nebola dostatočná. Nakoniec sme sa rozhodli ponechať dĺžku vstupného vektora na hodnote 48, pretože dosiahnutá presnosť je približne okolo hodnoty 1.6 a to aj pri relatívne malom počte neurónov na skrytej vrstve.

Vyhodnocovanie prihrávk sa vykonáva pomocou funkcie *getoutput(vektor)*, kde *vektor* je pole celých čísel s dĺžkou 48.

Neurónová sieť, ktorá je súčasťou prototypu je použiteľná iba na vyhodnocovanie za predpokladu, že má už načítané váhy zo súboru. Súborny neurónovej siete sú uložené v adresári *FM_Bla*. Okrem zdrojových súborov triedy *FM_Network* sa tam nachádzajú aj súborny pre triedu *Neuron*, ktorá implementuje jednotlivé neuróny v N.S.

5.2 PRIHRÁVKY

5.2.1 Zisťovanie úspešnosti prihrávk

Súčasný stav

Tím, z ktorého vychádzame, teda FC Farmári, riešia problematiku zisťovania prihrávk veľmi zvláštnym, dalo by sa povedať až pochybným spôsobom. Zisťovanie implementuje metóda *BehaveWithBall()* v triede *PlayerTactics*, ktorá je pre fázu tréningu rozšírená o algoritmus overovania úspešnosti prihrávk. Vytváranie súborov sa vykoná iba vtedy, ak je definované makro *PASS_TRAINING*. Na základe makra sú vytvorené určité moduly v kóde triedy *PlayerTactics*. Tréning začína tak, že sa nastaví v konštruktoze triedy premenná typu FILE *ff* na Null a boolean premenná *chytisom* na hodnotu false. Pri prvom použití metódy *BehaveWithBall()* sa na základe testovania hodnoty *ff* (ktorá je Null) vytvorí nový súbor, ktorého názov je nasledovný:

hracXY.txt, kde *X* je hodnota, ktorá špecifikuje príslušnosť hráča k tímu (1 pre tím na ľavej strane, 0 pre tím na pravej strane) a hodnota *Y* určuje číslo hráča. Súbor má nasledujúci formát:

Cas:TIME;Som:ID;POS_X;POS_Y;EVENT

Kde *TIME* je aktuálny čas, *ID* je identifikátor (číslo), *POS_X* a *POS_Y* udávajú polohu, *EVENT* je typ udalosti. Typy udalostí sú dokumentácie FC Farmári nasledovné:

- 0 – definícia seba samého, potom hodnota *ID* je vlastné číslo hráča, *POS_X* a *POS_Y* sú súradnice polohy seba samého.
- 1 – udalosť kopu do lopty, vtedy sa do *ID* zapíše číslo spoluhráča ktorému sa bude prihrávať a *POS_X,POS_Y* sú súradnice spoluhráčovej polohy
- 2 – definícia protihráča, potom hodnota *ID* je číslo protihráča spoluhráča a *POS_X, POS_Y* sú súradnice protihráčovej polohy
- 3 – udalosť chytenia lopty

Tréning sa nám bohužiaľ nepodarilo reprodukovať, v dôsledku nefunkčnosti dôležitej časti učiaceho mechanizmu, nástroja *HotParser*, ktorý zaznamenané výstupy prezentuje ako vstupy neurónovej siete, následne realizujúcej učiaci proces. Zo zdrojového kódu sa nám podarilo identifikovať nasledujúci algoritmus.:

1. Hráč vyhodnotí situáciu ako vhodnú príležitosť na nahrávku. Vyberie si adresáta a nahrá.

2. Do súboru zapíše svoju pozíciu, číslo a kód akcie, ktorú chce vykonať (v tomto prípade nahrávka), pozíciu adresáta a jeho číslo a pozície spolu s číslami všetkých súperov, ktorých vidí.
3. Nastaví *vlastnú* boolean premennú *chytlsom* na hodnotu true.
4. V ďalšom cykle na začiatku hráč zisťuje hodnotu premennej *chytlsom* a ak je rovná hodnote true, zapíše do súboru svoju pozíciu, číslo a kód úspešného prijatia prihrávky. Následne nastaví premennú *chytlsom* na hodnotu false a celý cyklus sa opakuje.

Už na prvý pohľad je jasné, že tu čosi nie je v poriadku. Konkrétne premenná *chytlsom*. Logika algoritmu naznačuje, že premenná predstavuje akýsi identifikátor pre hráča, ktorý má práve loptu, že mu ju nahral spoluhráč a teda, že sa jedná o úspešnú prihrávku. Hráči však existujú ako samostatné vlákna, pričom každý má svoju vlastnú premennú *chytlsom*, ktorá sa nastaví na hodnotu true len hráčovi realizujúcemu prihrávku. Je jasné, že v ďalšom cykle neoznamuje úspech adresát prihrávky, pretože jeho *chytlsom* má hodnotu false, ale ani prihrávajúci hráč, keďže už nemá loptu, teda nie je volaná metóda *BehaveWithBall()*. Ešte stále má však jeho chytlsom hodnotu true a teda v okamihu, keď loptu získa, oznámi úspešnú nahrávku, aj v prípade, ak sa k lopte dostal iným spôsobom (napr. získal ju od súpera).

Navrhované zmeny

Považujeme za nevyhnutné opraviť súčasný algoritmus zisťovania úspešnosti prihrávok, keďže sa jedná o kľúčový prvok pre správne fungovanie neurónovej siete. Ako najjednoduchší spôsob sa nám javí zmena premennej *chytlsom* z lokálnej premennej objektu na zdieľanú premennú všetkých procesov, predstavujúcich hráčov. Možné problémy, vyplývajúce z uvedeného riešenia môžu nastať napríklad v prípade, že hráč vykoná prihrávku, ale loptu zachytí súper. Ak by premenná ostala aj naďalej nastavená na hodnotu true, v momente, keď sa k lopte dostane náš hráč, bude ju pokladať za úspešnú nahrávku, čo je samozrejme chybné. Ošetrenie uvedeného prípadu spočíva v rozšírení metódy *BehaveWithoutBall()*, ktorá implementuje správanie sa hráča bez lopty, o nastavenie premennej *chytlsom* na false, ak vidí, že loptu má protihráč. Hoci hráči majú obmedzené videnie, je takmer na 100 percent jasné, že aspoň jeden z nich vidí protihráča s loptou a keďže premenná je zdieľaná, operácia splní svoj účel.

Uvedené riešenie môže predstavovať istý problém najmä z hľadiska prístupu k pamäti viacerými programami. Preto uvádzame alternatívu v podobe pozmeneného zápisu do výstupných súborov. Keďže prihrávajúci hráč vždy vie, komu ide nahráť (túto informáciu zapisuje vo forme čísla spoluhráča), nie je problém overiť, či sa k lopte adresát prihrávky naozaj dostal. Zápis do súborov sa rozšíri o zápis zo strany súperov, ktorí však budú zapisovať len skutočnosť, že sa dostali k lopte. Parser úspešnosť prihrávky vyhodnotí na základe dvoch po sebe idúcich zápisov súbore a času, v ktorom sa k lopte dostal súper, pričom úspech je podmienený skutočnosťou, že adresát prihrávky v čase 1 prihráva v čase 2 a v tomto rozmedzí sa k lopte nedostal súper.

Ďalšou nutnou zmenou, ktorú musíme vykonať je implementácia parsera na prevod výsledkov zapísaných do výstupných súborov na vstupy neurónovej siete. Spomínaný

HotParser, ktorý implementoval predchádzajúci tím sa ukázal ako nefunkčný, dokonca neskompilovateľný v prostredí Visual C++. Keďže sa jedná o dôležitú súčasť celkového riešenia, budeme mu venovať vysokú prioritu na začiatku letného semestra. Pri vytváraní tréningových vzoriek pre použitie viacerých neurónových sietí sme predpokladali, že jeden tím bude mať pevne stanovenú hlavnú formáciu a druhý sa si bude tieto formácie ľubovoľne meniť. Preto navrhujeme vytvoriť dva typy tímov, kde v tíme, ktorý bude trénovať prihrávky sa budú môcť meniť formácie a jeho hráči si budú navzájom prihrávať. Formát súborových výstupov hráčov tímu bude rovnaký ako majú FC Farmári. Premenná *chytilsom* v súčasnej podobe bude z kódu hráča odstránená. V prípade, že sa hráč dostane akýmkoľvek spôsobom k lopte zápis do súboru nasledovný:

- 3 – udalosť chytenia lopty
- 0 – definícia samého seba
- 1 – udalosť kopu do lopty

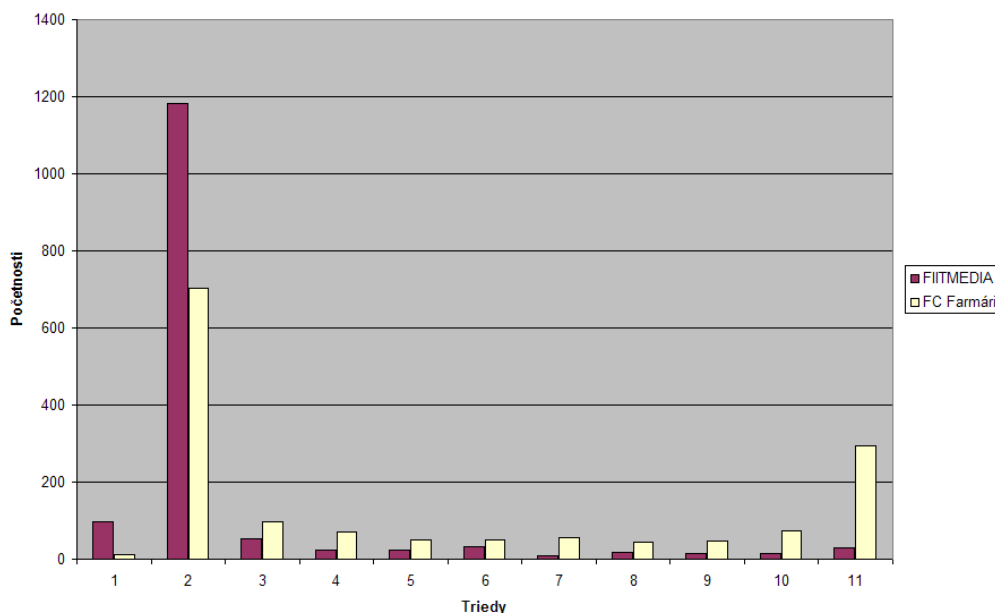
v cykle pre všetkých protihráčov

- 2 – definícia protihráča

Hneď za týmto zápisom sa uskutoční prihrávka spoluhráčovi.

Druhý tím s pevnou základnou formáciou, bude do súboru zapisovať iba udalosti chytenia lopty – 3. Premenná *chytilsom* bude taktiež z kódu hráča odstránená. Hráči budú môcť driblovať a kopat' na bránu, čím sa docielia, že hráči dokážu preniknúť hlbšie do súperovej obrany.

Navrhnutým tréningom sa síce zníži počet tréningových vstupov na približne polovicu, ale na druhú stranu sa podstatne uľahčí implementácia parsera súborov.



Obr. 5-1 Histogram vyhodnocovania úspešnosti prihrávok neurónovou sieťou

Na základe dostupných výstupných dát od FC Farmárov sme na vyhodnocovanie úspešnosti prihrávok vyskúšali v praxi pôvodnú neurónovú sieť FC Farmárov a našu

neurónovú sieť. Výsledky sme spracovali a vytvorili histogram, zachytený na Obr. 5-1, v ktorom sú znázornené jednotlivé triedy (interval 0 – 1, s krokom 0,1, pričom 0 predstavuje neúspešnú a 1 úspešnú prihrávku) a k nim prislúchajúce početnosti (vzorka 1497 údajov). Z histogramu je zrejme, že tréningová množina nie je na spoľahlivé natréningovanie siete vhodná, keďže väčšina hodnôt sa pohybuje okolo triedy 2, ktorá predstavuje veľmi nízku úspešnosť prihrávky a teda nízku pravdepodobnosť jej realizácie. V prípade našej siete dopadli výsledky ešte horšie, zrejme v dôsledky prísnejšieho mechanizmu vyhodnocovania.

Dôsledkom experimentu je teda potreba venovať zvýšené úsilie výberu vhodnej tréningovej množiny, čo sa aj pokúsime realizovať.

5.2.2 Prihrávky do behu a obsadeným hráčom

Súčasný stav

V súčasnosti sa prihrávky do behu v rámci hráča tímu FC Farmári vôbec nevyužívajú, hoci v zdrojovom kóde sa metóda, ktorá ich implementuje nachádza. Jedná sa o metódu *PassToRun()*. V rámci prototypu sme sa rozhodli metódu vyskúšať a zakomponovali sme ju do rozhodovania sa hráča. Výsledky nie sú príliš optimistické. Výrazne sa zvýšila nepresnosť nahrávok a navyše sa nedostavil očakávaný nárast ofenzívnej sily, keďže lopty väčšinou končia u spoluhráčov. Je evidentné, že algoritmus bude treba prepracovať. Autori v rámci určovania sily prihrávky počítajú s maximálnou rýchlosťou pohybu hráča, avšak nijakým spôsobom adresáta prihrávky na to, že má bežať maximálnou rýchlosťou neupozorňujú. Aj to by mohla byť príčina neúspechu.

Navrhované zmeny

Pre zvýšenie úspešnosti prihrávok do behu sme sa rozhodli implementovať aj posielanie správy adresátovi prihrávky, ktorá ho upozorní, aby bežal určenou (spravidla maximálnou) rýchlosťou na želané miesto, kde by mal prihrávku prebrať. Pri vytváraní správy sa budeme inšpirovať tímom Squirrel Squadron, ktorý sa na vytváranie a posielanie správ úzko špecializoval minulý rok.

Ďalšia zmena súvisí s realizáciou prihrávok obsadeným hráčom, ktoré sa vyskytli už v časti návrh, ale kvôli nedostatku času sme sa im zatiaľ nestihli venovať. V podstate sa jedná o podobný princíp ako prihrávky do behu s tým, že hráč sa rozhoduje v špecifickej situácii. Taktiež bude potrebné implementovať správu, oznamujúcu adresátovi prihrávky ako rýchlo a kam má bežať. V záujme momentu prekvapenia sa bude jednať o maximálnu možnú rýchlosť.

Pokúsime sa implementovať správy stručne a svojím obsahom výstižne a obmedziť ich počet na minimum, aby sme sa vyhli zahlteniu a následným „uviaznutiam“, ktoré boli tak časté u tímov, ktoré sa posielaním správ zaoberali.

5.3 Kouč

Kouč tímu FC Farmári obsahuje množstvo tried, ktoré nie sú dobre zdokumentované, komentovanie zdrojových súborov je na dosť slabej úrovni.

Zistili sme, že kouč tímu FC Farmári je nestabilný – často krát počas zápasu vyvolá neošetrenú výnimku a predčasne sa ukončí. Niekedy „pád“ programu kouča spôsobí aj „pád“ programov ostatných agentov – t.j. hráčov. Túto chybu nepozorovali rovnomerne všetci členovia tímu. Najviac k nej dochádzalo pod operačným systémom Microsoft Windows 2000.

Namiesto predsavzatej analýzy vlastností kouča a ich vylepšovania sme sa snažili prioritne odhaliť opísanú chybu, pretože jej výskyt v zápase môže znamenať kontumačné ukončenie zápasu v náš neprospech, ak by došlo aj k následnému „pádu“ hráčov a zápas by sa musel zrušiť. Chybu sa nám odhaliť zatiaľ nepodarilo. Zatiaľ vieme, že k chybe dochádza pri zapisovaní do pamäte, ktorá programu nepatrí. Ako možné príčiny sme identifikovali problémy s východiskovou formáciou, prípadne čítanie z chybného konfiguračného súboru, ktoré môže potenciálne spôsobiť prácu s nekorektnými údajmi. Odhaľovaniu chyby sa plánujeme ďalej venovať v letnom semestri.

5.4 VIZUALIZÁCIA ROZHODOVACIEHO STROMU

Pri analýze hráča tímu FC Farmári a pri následnej práci na prototypu sme dospeli k záveru, že jeho taktická vrstva je príliš komplikovaná, rozhodnutia hráča je ťažké vysledovať a zdôvodniť a prijaté rozhodnutia hráča sú často nesprávne (ide hlavne o rozhodnutie, čo robiť s loptou, keď neurónová sieť na prihrávky neodporučí vykonanie konkrétnej prihrávky, o techniku driblingu a rozhodovanie sa hráčov pri štandardných situáciách).

Rozhodli sme sa v letnom semestri vykonať potrebné úpravy v taktike hráča. Aby sme to mohli úspešne dosiahnuť, musíme najprv pochopiť jeho terajšie rozhodovanie a byť schopní navrhnúť lepšie alternatívy. Na uľahčenie dosiahnutia tohto cieľa sme sa rozhodli vizualizovať herný strom (ide vlastne o graf) hráča tímu FC Farmári. Pritom sme vychádzali z podobnej vizualizácie hráča tímu Sklo vykonanej tímom Squirrel Squadron.

Tím Squirrel Squadron vytvoril grafickú reprezentáciu kódu hráča pomocou vývojových diagramov v nástroji Microsoft Visio. Týmto spôsobom sa im podarilo odhaliť a opraviť niektoré chyby implementácie pôvodného hráča.

Grafická reprezentácia rozhodovacieho stromu tímu Sklo je dosť rozsiahla – predstavuje ju 26 vývojových diagramov. My sme sa rozhodli vizualizovať len najpodstatnejšiu metódu hráča tímu FC Farmári – metódu *PlayerTactics::BehaveWithBall()*, ktorej pochopenie je náročné, pretože táto metóda je veľmi rozsiahla (cez 250 riadkov kódu v jazyku C++) a obsahuje previazané časti kódu pre hru s kódom pre tréning. Pri vizualizácii sme prevzali pôvodnú šablónu tímu Squirrel Squadron *CCF_stencil.vsx*.

Výsledky vizualizácie metódy *PlayerTactics::BehaveWithBall()* sú uvedené na obrázkoch v prílohe H. Plánujeme ich ďalej využiť v letnom semestri pri zmenách taktickej vrstvy.

5.5 LIBERO

Časť návrhu a špecifikácie týkajúca sa libera a využitia štatistickej analýzy rozloženia súperových hráčov pre potreby libera nebola v prototypy riešená z dôvodu výskytu úloh s vyššou prioritou (objavenie chyby v koučovi a zistenie nedostatkov v taktickej vrstve hráča). Plánujeme sa jej venovať v letnom semestri podľa časových možností.

5.6 OVERENIE POUŽITIA VZOROV A MASIEK

V tejto časti sa budeme venovať prácam súvisiacim s overením možnosti použitia vzorov a masiek z pohľadu pamäťovej a časovej zložitosti.

Prvým cieľom bolo rozpoznanie miesta, ktoré by bolo vhodné na volanie navrhovanej funkcionality. Po dlhšej analýze existujúceho kódu sme pre vhodné miesto vybrali funkciu *BehaveWithBall()*, ktorá je metódou triedy *PlayerTactics*. Je to z toho dôvodu, že použitie vzorov je opodstatnené len pri hre s loptou.

Túto triedu sme ďalej rozšírili o pole masiek. Pôvodne sme uvažovali o aplikovaní vzorov na dvoch spoluhráčov, v dôsledku čoho sa pamäťové nároky kvadraticky zvyšujú vzhľadom na aplikovanú mriežku. Ako minimálny počet riadkov/stĺpcov mriežky sme určili číslo 15 a minimálny rozmer jednej bunky na 1 meter. Z toho vyplýva, že počet buniek mriežky bude 225 a počet potrebných miest pre masky by bol 50625.

V konštruktore triedy sme vyhradili miesto pre masky a potom skúsili spustiť simuláciu zápasu. Nastali problémy s inicializáciou hráčov a druhý tím sa prakticky nepodarilo rozbehnúť. Za takýchto podmienok nie je možné pracovať na ďalších zlepšeniach a tak sme možnosť aplikovania vzorov na dvoch spoluhráčov vylúčili.

Ďalej sme určili maximálne obmedzenie pre mriežku. Vzhľadom na obmedzenie viditeľnosti, určili sme maximálne rozmery na 30x30 metrov, pri rozostupe 1 meter. Takto potrebujeme maximálne 900 masiek, ak k jednému vzoru prislúcha jedna maska. Po upravení hráča sme spustili simuláciu. Pridanie masiek sa neprejavilo na správaní hráča.

Keďže pamäťové nároky len čiastočne obmedzujú pôvodný návrh, pokračovali sme v otestovaní časovej zložitosti. Bol navrhnutý nasledujúci algoritmus:

1. Pre každého spoluhráča sa vyhodnotí jeho pozícia v mriežke relatívne k pozícií hráča s loptou. Každé políčko je očíslované od 1 do 900.
2. Podľa tohto čísla sa vyberie maska.
3. Každého protihráča otestujeme, či sa nenachádza v mriežke, a určíme jeho polohu. Maska je reprezentovaná 30 hodnotami typu integer (32 bitov).
4. Z polohy protihráča vypočítame pozíciu v maske. Z y-ovej osi zistíme, v ktorom čísle sa nachádzajú prislúchajúce bity pre daný riadok a z x-ovej pozíciu konkrétneho bitu prislúchajúceho pozícií v maske a pomocou bitových operátorov zistíme, či sa daný protihráč nachádza na bezpečnej pozícií alebo nie. Ako bezpečnú pozíciu označíme 1, inak 0.

Pred a po otestovaní pomocou mriežky sa vypíše čas v milisekundách. Po spustení simulácie a porovnaní zobrazených hodnôt môžeme konštatovať, že čas potrebný na rozpoznanie vzorov a masiek nie je merateľný a preto je použitie vzorov a masiek v praxi použiteľné.

Pre generovanie masiek sa vytvorí program, ktorý bude fungovať tak, že pre každú polohu protihráča v mriežke sa vypočíta, či je schopný prekaziť prihrávku. Ak z tejto pozície nie je možné prekaziť prihrávku, potom je protihráč v bezpečnej oblasti.

Zhodnotenie

Z výsledkov vyplýva, že takéto riešenie je možné použiť bez zvýšených nárokov na pamäť alebo čas potrebný na ohodnotenie situácie. Takto definované použitie vzorov a masiek je obdobou použitia neurónových sietí. Na rozdiel od nich však ide o deterministické vyhodnotenie situácie. Keďže hráč je už vo veľkej miere prispôsobený na použitie neurónových sietí, prioritne sa budeme venovať neurónovým sieťam.

5.7 ZHODNOTENIE PROTOTYPU

Hráč tímu FC Farmári je implementovaný neprehľadne a vyskytujú sa tam nielen logické ale aj implementačné chyby. Napriek tomu sme sa rozhodli pokračovať v jeho zlepšovaní, keďže z funkcionálnej stránky tvorí dobrý základ. Prevažnú väčšinu chýb sa nám podarilo lokalizovať a pokúsime sa ich odstrániť.

Vzhľadom na existujúce problémy v pôvodnom hráčovi, pridali sme navrhovaným úlohám priority. Prvé tri priority obsahujú úlohy, ktoré je potrebné bezpodmienečne vykonať pred nasledujúcimi prácami, pretože sú to opravy existujúcich chýb v pôvodnom hráčovi.

Hlavnou prioritou v oblasti implementácie novej funkcionality bude vylepšenie rozhodovania pomocou neurónových sietí a prihrávkov hráčov. Libero a vzory budeme implementovať podľa časových možností.

Priorita I

- oprava kopania,
- oprava chyby spôsobujúcej pád kouča ,

Priorita II

- oprava správania sa hráča po nerozhodnutí neurónky (funkcia *BehaveWithBall()*),
- odstránenie statickosti hráča pri hre bez lopty (funkcia *BehaveWithoutBall()*),
- použitie kričania pre posielanie správ (prihrávky do behu),

Priorita III

- implementácia neurónových sietí,

Priorita IV

- implementácia Libera,

Priorita V

- implementácia vzorov a masiek

6 OPIS RIEŠENIA

Táto kapitola obsahuje opis riešenia projektu. Jej obsahom sú výsledky našej práce dosiahnuté počas implementácie riešení navrhnutých v návrhu a overovaných v prototypu. Každému riešenému problému je venovaná samostatná podkapitola.

6.1 ÚPRAVA ALGORITMU NA PRIHRÁVANIE

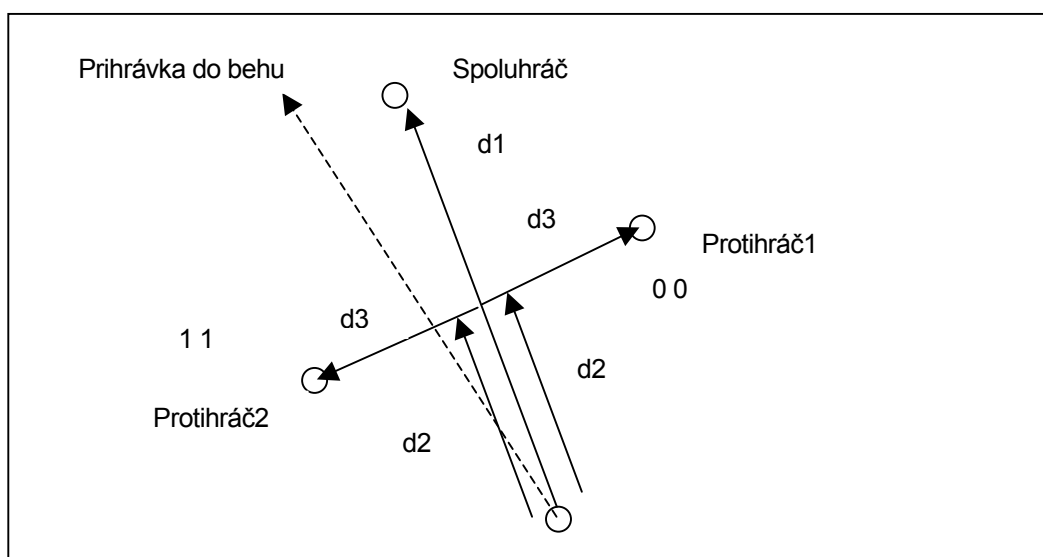
Počas prehliadky kódu hráča FC Farmári sme zistili, že hráč pri prihrávkach automaticky používa prihrávky do behu. Po analýze hry sme sa rozhodli tento spôsob zmeniť, pretože niekedy nie je možné takúto prihrávku realizovať bez toho, aby ju zachytil súper. Na druhej strane sú však prihrávky do behu často výhodne, pretože urýchľujú hru. Preto sme sa rozhodli rozlišovať dva druhy prihrávok, a to prihrávky na miesto a prihrávky do behu.

Prihrávky do behu sú implementované pomocou odladenej konštanty, ktorá sa vynásobí vzdialenosťou k hráčovi, ktorému bude prihrávať a následne sa tento súčin pripočíta k x-ovej súradnici tohto hráča. Bod, kde sa bude prihrávať možno zapísať ako

$$[\text{spoluhrač}.X+0,2*\text{dist}, \text{spoluhrač}.Y]$$

Aby dokázala neurónová sieť rozlíšiť, či ide o prihrávku do behu alebo na miesto, pridali sme do vektora dva ďalšie bity: 00 pre prihrávky na miesto a 11 pre prihrávku do behu.

S prihrávkami do behu sa vyskytol aj ďalší problém, ktorý súvisel s pôvodnou štruktúrou dát, ktorá nezohľadňovala, z ktorej strany sa príslušný protihráč nachádza. Preto sme sa rozhodli pridať do vstupu neurónovej siete ďalšie dva bity, ktoré to zohľadňujú. Na nasledujúcom obrázku je možné vidieť, aký je rozdiel v situácii, kedy budú mať dvaja protihráči rovnaké relatívne vzdialenosti, ale nachádzajú sa na opačných „stranách“ prihrávky.



Obr. 6-1 Situácia spoluhráč a dvaja protihráči. V pôvodnom formáte mali protihráči rovnaký vstupný vektor, teraz sú odlišení pomocou dvoch bitov, pravá strana - protihráča1 dva bity 0 0, ľavá strana - protihráča2 dva bity 1 1.

Hlavný dôvod spočíval v tom, že protihráč2 ohrozuje prihrávku do behu oveľa viac ako protihráč1. Celkový vektor, ktorý vstupuje do neurónovej siete je nasledujúceho formátu

d1 d2 d3 štýl strana výsledok

kde štýl je štýl kopania:

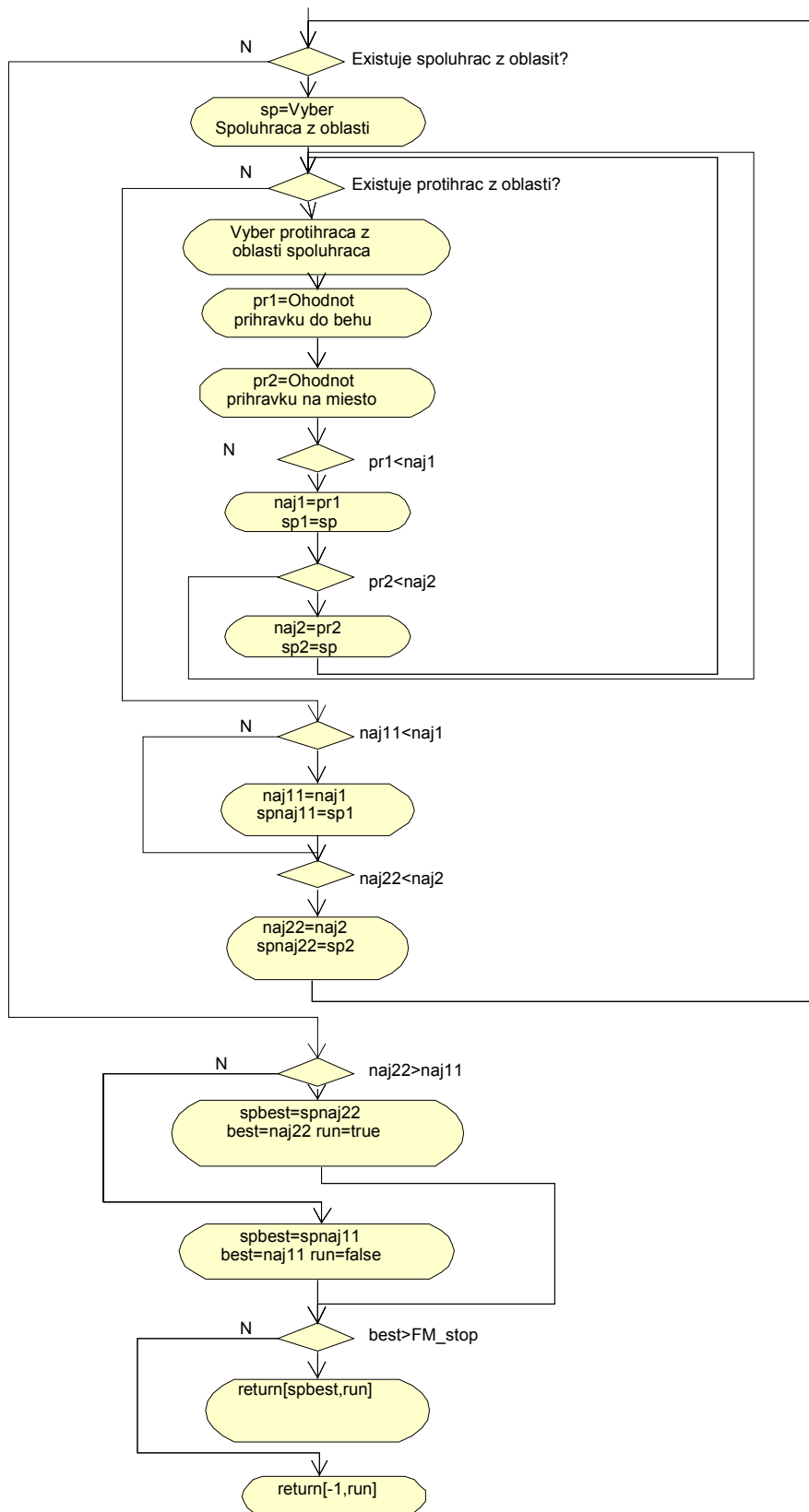
- 0 0 na miesto
- 1 1 do behu

a strana je strana z ktorej je súper:

- 0 0 sprava
- 1 1 zľava

Vzhľadom na to, že neurónová sieť má rozšírený vstup bolo nutné zmeniť aj algoritmus výberu hráča, ktorému sa bude prihrávať. To znamenalo ďalšiu úpravu funkcie `FCGetPassAbleFriend`. Algoritmus pre každú situáciu spoluhráč - protihráč vyhodnotí prihrávku na miesto a do behu. Pre každého spoluhráča hľadá protihráča, ktorý ho ohrozuje najviac (pravdepodobnosť prihrávky je najmenšia). Rôzni protihráči ohrozujú spoluhráča inak v prihrávke do behu a inak v prihrávke na miesto. Následne sa hľadá maximálna pravdepodobnosť spomedzi všetkých spoluhráčov (avšak separátne pre prihrávky do behu a pre prihrávky na miesto). Na záver sa určí, či sa bude prihrávať do behu alebo na miesto porovnaním týchto dvoch najvyšších pravdepodobností. Pravdepodobnosť prihrávky sa porovná s hraničnou hodnotou `FM_stop`. V prípade, že je väčšia, tak funkcia vráti číslo hráča aj s príslušným štýlom kopania (uloženým v boolovskej premennej `run`). Ak je menšia, tak sa vráti hodnota -1.

Zjednodušený algoritmus tohto prehľadávania je na Obr 6-2. Premenné, ktoré sú na obrázku, nie sú z dôvodu čitateľnosti rovnaké ako sa používajú vo funkcii `FCGetPassAbleFriend`.



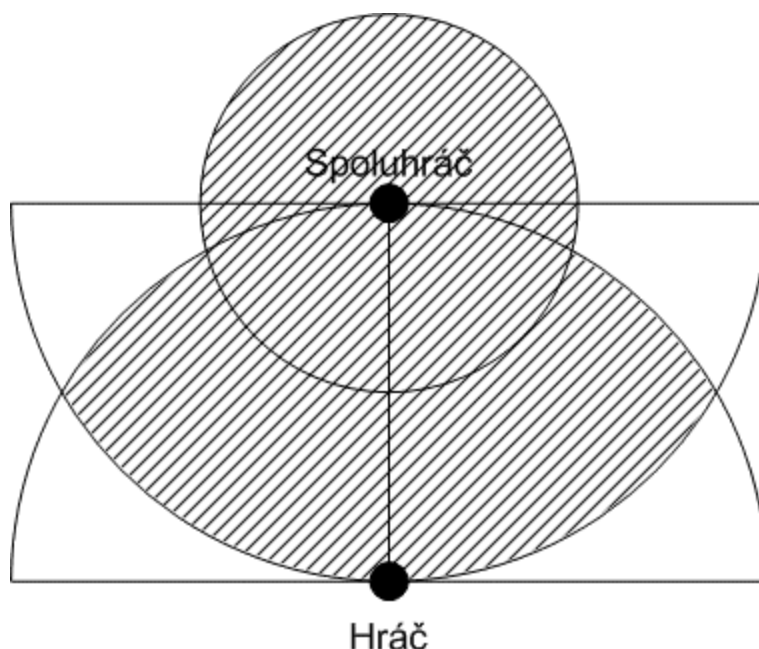
Obr. 6-2 Algoritmus výberu spoluhráča

Samotná funkcia `FCGetPassAbleFriend` je ešte o niečo zložitejšia, pretože okrem algoritmu uvedeného na obrázku ešte v prípade, že existuje viac najlepších riešení, (ak

neurónová sieť nemala čo vyhodnocovať, pretože žiaden protihráč nebol v definovanej oblasti, tak pravdepodobnosť prihrávky je implicitne 1), tak nájde prislúchajúceho spoluhráča, ktorý je najbližšie k súperovej bráne. V hlavnom cykle funkcie `BehaveWithBall` sa už nepoužíva funkcia `GetPassAbleFriend`, ale rozšírená a upravená funkcia `FCGetPassAbleFriend`.

6.2 OKOLIE OHROZENIA SPOLUHRÁČA

Protihráči, ktorých vzdialenosti vstupujú do vektora neurónovej siete nie sú vyberaní náhodne. To, či protihráč bude do okolia ohrozenia hráča patriť, sa rozhodne podľa výsledku boolovskej funkcie `FCBelongsToArea(Point A, Point B, Point E)`, ktorá za ohrozujúcich určí protihráčov nachádzajúcich sa vo vyšrafovannej oblasti na Obr. 6-3. Polomer kruhu okolo spoluhráča, ktorému hráč nahráva, je rovný polovici ich vzdialenosti.



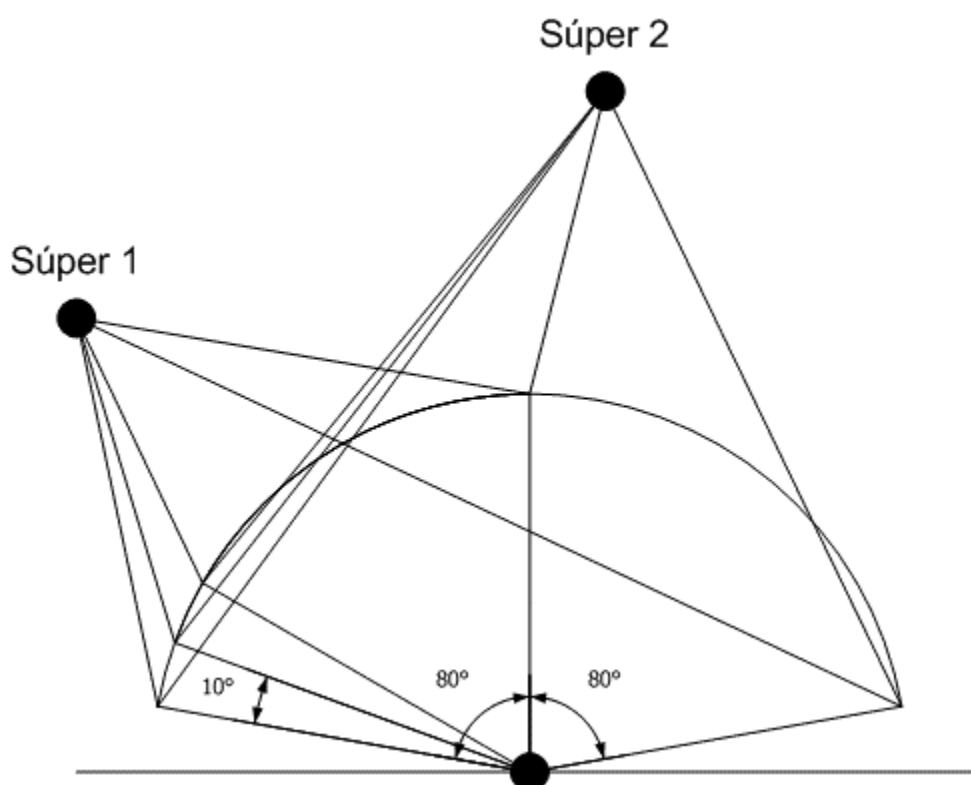
Obr. 6-3 Okolie ohrozenia spoluhráča

6.3 NAKOPÁVANÝ DRIBBLIG

Počas testovania hráča FC Farmári sme zistili, že hráč nedokáže počas driblovania podržať loptu dostatočne dlho na to, aby dokázal naviazať na seba dostatočný počet súperových hráčov. Ďalší nedostatok pôvodného driblingu spočíval v jeho rýchlosti, pretože ak má hráč pomerne veľa priestoru pred sebou, tak je vhodné, aby si loptu nakopol čo naviac dopredu. Z tohto dôvodu sme sa rozhodli implementovať tzv. nakopávaný dribbling, pri ktorom si hráč v prípade voľného priestoru dokáže loptu nakopnúť do väčšej vzdialenosti. Nakopávaný dribbling je implementovaný vo funkcii `FM_DribbleTo(const Point& pos)` triedy `PlayerSkills`. Funkcia rozhoduje ako a kam sa bude driblovať, na základe nasledovného algoritmu: Priestor okolo spojnice k cieľu sa rozdelí na body, ktoré

sú vzdialené od hráča o rovnakú vzdialenosť (akýsi polomer), ale každý z nich je od osi spojnice otočený pod iným uhlom. Uhly od spojnice k cieľu v stupňoch a sú nasledovné: -80° , -70° , -60° , -50° , -40° , -30° , -20° , -10° , 0° , 10° , 20° , 30° , 40° , 50° , 60° , 70° a 80° . Funkcia pre každý bod prislúchajúci k danému uhlu určí najmenšiu vzdialenosť protihráča z pomedzi všetkých protihráčov. Vo všeobecnosti, každý takýto bod má túto najmenšiu vzdialenosť inú a môže patriť k rôznym protihráčom (ktorému protihráčovi patrí, je irelevantné).

Hráč vyberie bod s najväčšou vzdialenosťou a vynásobí jeho vzdialenosť s odladenou konštantou FM_Mul . Túto hodnotu porovná so svojou vzdialenosťou (polomerom). Ak je táto vzdialenosť väčšia, tak hráč kopne loptu do daného bodu a nastaví hodnotu FM_TIME v závislosti od polomeru. (Hodnota FM_TIME je potrebná na to, aby bol hráč nútený bežať za loptou a tak pokračovať ďalej v driblingu). Ak je táto vzdialenosť menšia, tak sa zníži polomer a vyhodnocovanie sa začne odznova. Aby hráč nekopal loptu stále do krajných uhlov, tak jednotlivé vzdialenosti všetkých uhlov rôznych od 0° sú penalizované (vzdialenosti sa „zmenšujú“ a protihráč sa javí ako by bol bližšie) pomocou odladenej funkcie $FM_AnglePenalty$. Ak polomer klesne pod určitú hodnotu, hráč prejde na obyčajný dribbling, ktorý je implementovaný v pôvodnom hráčovi tímu FC Farmári. Situáciu ilustruje obrázok 6-4.



Obr. 6-4 Ilustrácia myšlienky nakopávaného driblingu

6.4 PROBLÉMY S KOPANÍM

Pri analýze prihrávkov sme zistili, že hráč často kope loptu iným smerom (a zároveň do iného bodu) ako má. Niekedy je tento rozdiel veľmi veľký. Pri analýze presnosti kopania sme zistili, že najväčší vplyv na presnosť majú nasledujúce faktory:

- Veľkosť rýchlosti lopty a veľkosť rýchlosti s akou má lopta prísť do cieľa
- Uhol medzi vektorom hráča a vektorom rýchlosti lopty
- Uhol medzi vektorom hráča a vektorom cieľa
- Uhol medzi vektorom hráča a vektorom rýchlosti lopty
- Uhol medzi vektorom cieľa a vektorom rýchlosti lopty

Keď sa hráč rozhodne kopat' (prihrávať, strelať na bránu a pod.), tak sa samotné kopanie vykoná tak, že sa zavolá funkcia `PassBallTo(target, req_spd)`, kde `target` je bod reprezentujúci cieľ a `req_spd` je veľkosť rýchlosti, s ktorou má lopta prísť do cieľa. Vo funkcii `PassBallTo` sa vykoná nasledujúci výpočet, ktorý musí spĺňať dve rovnice:

$$spd_{REQ} = spd_{ZAC} \times (balldecay)^n$$

$$dist = \sum_{i=0}^{n-1} spd_{ZAC} \times (balldecay)^i = spd_{ZAC} \times \frac{1 - balldecay^n}{1 - balldecay}$$

kde `dist` je vzdialenosť lopty od cieľa. V rovniciach sú dve neznáme:

- `n` - počet cyklov, za ktoré sa lopta dostane do cieľa
- `spdZAC` - počiatočná rýchlosť lopty, ktorú musí lopta mať aby sa dostala do cieľa

Vyjadrením neznámej `n` z prvej a následným dosadením do druhej rovnice dostávame

$$dist = \frac{spd_{ZAC} - spd_{REQ}}{1 - balldecay}$$

Následným vyjadrením `spdZAC` dostávame finálnu veľkosť rýchlosti počiatočnej

$$spd_{ZAC} = dist \times (1 - balldecay) + spd_{REQ}$$

Využitím `spdZAC` sa vytvorí vektor `v` smere cieľa (vektor dĺžky `spdZAC`) a ten sa pripočíta k pozícii lopty. Výsledný bod sa odošle na ďalšie spracovanie funkcii `NearKick(target)`. Vo funkcii `NearKick` sa potom počíta potrebná sila kopu vzhľadom na rozdiel uhlov medzi vektorom hráča a vektorom lopty. Do výpočtov sa započíta aj efektívna sila kopu hráča, ktorá závisí od vzájomnej polohy hráča (uhla a vzdialenosti) nasledovne:

$$ep = ep \left(1 - 0.25 \frac{dir_diff}{180} - 0.25 \frac{dist_ball}{kickable_margin} \right)$$

Kde `ep` (effective power) sa vypočíta nasledovne

$$ep = kick_power \times kick_power_rate$$

Ako je vidieť, tak efektívna sila kopu môže klesnúť až na polovicu a to v závislosti od vzájomnej polohy hráča a lopty.

Aby sa odstránila nežiadúca časť vektora rýchlosti lopty, musí hráč kopnúť do lopty tak, aby išla požadovaným smerom bez ohľadu na to, aký má smer rýchlosti v tejto chvíli. Nasleduje výpočet zrýchlenia, ktoré musí byť lopte dané, aby lopta prišla do cieľa požadovaným spôsobom. Ak je toto zrýchlenie väčšie ako je maximálne možné zrýchlenie lopty (vypočítané podľa hore uvedeného vzorca pre ep), tak nasleduje výpočet pomocou kvadratickej rovnice, ktorému sa nám však nepodarilo porozumieť. Vydedukovali sme však, že hráč sa pokúša kopnúť do lopty tak, aby sa pohybovala približne požadovaným smerom a rýchlosťou. Pokúšali sme sa preto tento spôsob vyhodnocovania zmeniť tak, že ak nie je možné loptu kopnúť do požadovaného bodu a smeru, potom hráč loptu zastaví (kopne do lopty opačným smerom). Bohužiaľ počas hry sme zistili, že hráč veľmi často nemôže kopnúť požadovaným spôsobom a musí loptu neustále zastavovať. To má za následok rapídne spomalenie hry. Odporúčali by sme túto funkciu prerobiť, aby mala ako parameter aj požadovanú presnosť. Ak nebude možné kopnúť loptu s požadovanou presnosťou, tak potom sa musí lopta „zastaviť“.

Poznámka: Pri hre sme si všimli, že pokiaľ lopta ide na hráča pod príliš tupým uhlom, tak ju hráč spracuje (odkopne alebo zastaví) úplne zle a loptu urýchli v jej pôvodnom smere, namiesto toho, aby ju spomalil.

6.5 TRÉNOVANIE PRIHRÁVOK

Kapitola sa venuje problematike tréningu prihrávok. Vysvetľuje dôvody na tréningu prihrávok, stratégiu hry počas tréningu a jej dôsledky. Podrobne opisuje zber údajov o zápase a ich reprezentáciu, ako aj úskalia, s ktorými sme sa stretli. Nakoniec uvádzame rôzne alternatívy k výberu vhodných údajov na tréningu a ich zhodnotenie.

6.5.1 Dôvody tréningu

Základnou požiadavkou úspešnej hry sú vhodné a presné prihrávky. Dôležité je nielen, či sa prihrávka dostane k svojmu adresátovi, ale aj výber samotného adresáta. K problému sa dá pristupovať systematicky, na základe vnemov z okolia a ich vyhodnotenia s použitím pevne stanovených kritérií. Bohužiaľ, informácie, ktoré sa dostávajú k hráčovi so sebou nesú istú dávku nepresnosti, čo negatívne ovplyvňuje relevantnosť týchto kritérií. Ďalším problémom je množstvo rôznych situácií, pre ktoré je potrebné stanoviť rozhodovacie kritériá. Naopak, rozhodovanie pomocou neurónovej siete nepozná konkrétne kritériá, funguje na základe množiny vstupov, na ktoré je sieť natrénovaná. Znamená to, že hráč sa môže a nemusí rozhodnúť správne, závisí to od miery natrénovania siete. Táto neurčitost' predstavuje akési zadné vrátka pre prípady, ktoré sa nedajú ošetriť bežným stanovením rozhodovacích kritérií.

V praxi sme sa rozhodli do istej miery vyskúšať aj systematický prístup a porovnať ho s prístupom tréningu.

Systematický prístup

Dlhodobejším pozorovaním správania sa hráčov počas zápasu sme zistili niekoľko dôležitých skutočností týkajúcich sa prihrávok.

- Hráči majú tendenciu prihrávať spoluhráčom, ktorí sú relatívne blízko. Vzdialenejší hráči sa teda do hry zapájajú menej a navyše sa znižuje šanca preniknúť do súperovho obranného pásma.
- Hráči zvyknú prihrať spoluhráčovi vzadu väčšinou v momente, keď by bolo vhodnejšie driblovať alebo nahráť dopredu.
- Hráči často prihrávajú spoluhráčom, ktorí sú obsadení protihráčom, čím sa výrazne zvyšuje množstvo neúspešných prihrávok
- Posledná vec sa týka implementácie nižších funkcií hráča. Kopanie sa nie vždy správa podľa očakávaní, nedá sa predvídať.

Na základe zistených skutočností sme postupne implementovali nasledujúce prvky:

- Prihráva sa zásadne dopredu, čím sa hra prenáša bližšie k súperovej bráne.
- Prihráva sa vždy najvzdialenejšiemu hráčovi, takže sa využíva väčšia časť ihriska a súperovi hráči sa musia viac hýbať, vznikajú diery v obrane.
- Najvhodnejší adresáti prihrávok sú tí hráči, ktorí sú najmenej obsadení, čo má za následok nárast počtu úspešných prihrávok.

Postupne sme experimentovali s ďalšími prvkami, ako rozdielne správanie sa hráča v rôznych častiach hracej plochy, povolenie aj prihrávok dozadu. Zaznamenali sme čiastočné úspechy voči iným tímom, avšak z celkového pohľadu sme nedospeli k uspokojivému výsledku. Bolo skoro nemožné odladiť a zosúladiť jednotlivé rozhodovacie kritériá, ktoré často navzájom pôsobili kontraproduktívne. Odstránenie problémov v jednej časti viedlo k prehĺbeniu problémov v inej časti.

Všetky tieto skutočnosti nás utvrdili v presvedčení trénovať prihrávky pomocou neurónovej siete.

6.5.2 Tréning

Tréningom sa rozumie odohranie zápasov spolu so zberom údajov a spôsobom hry. Tréning neurónovej siete nasleduje až po zozbieraní údajov a ich následnej transformácii na vstupy. Transformácia a tréning siete sú opísané v inej časti dokumentu.

Stratégia hry

Stratégia hry počas trénovania prešla viacerými zmenami. Najskôr sme využili pôvodný prístup tímu FC Farmári, ktorí spočíval v prihrávaní vždy len najbližšiemu hráčovi. Nevhodnosť prístupu tkvie v nedostatočnom pokrytí rôznych situácií počas hry.

Riešením bolo prihrávanie náhodnému hráčovi z množiny viditeľných hráčov. Pokrytie hry už bolo takmer dostačujúce, avšak prístup v sebe niesol problém najmä v prihrávkach dozadu, čo nám jednak komplikovalo určovanie smeru protihráča a na druhej strane bránilo ofenzívnejšej hre.

Nakoniec sme zvolili nasledovnú stratégiu. Adresát prihrávky sa určuje náhodne, avšak len zo spoluhráčov, ktorí sa nachádzajú z pohľadu prihrávajúceho hráča vpredu. Vytvorili sme predpoklady na ofenzívnu hru, ktorá je kľúčom k úspechu. Úplne sme eliminovali prihrávanie dozadu, ktoré však samotnej hre často viac škodilo ako prospievalo.

Zvolenú stratégiu sme overili v procese tréningu a výsledky nás príjemne prekvapili. Nielenže hra získala na dynamike, ale dokázali sme hrať vyrovnanú partiu a viackrát aj poraziť tím FC Farmári, ktorý už prihrávanie natrénované mal.

Zber a reprezentácia údajov

Vhodný výber a reprezentácia vstupov je kľúčovým rozhodnutím pre efektívne fungovanie neurónovej siete. Po dôkladnom zvážení sme akceptovali tri typy údajov od tímu FC Farmári, ktoré sme rozšírili o ďalšie dva údaje, podľa nás upresňujúce hernú situáciu. Dokopy zaznamenávame týchto päť údajov:

- 1) Vzdialenosť medzi prihrávajúcim hráčom a adresátom prihrávky
- 2) Veľkosť priemetu vektora prihrávajúceho hráča a protihráča do vektora prihrávajúceho hráča a adresáta prihrávky
- 3) Vzdialenosť protihráča od vektora prihrávajúceho hráča a adresáta prihrávky
- 4) Typ prihrávky. Tj. buď prihrávka na telo alebo do behu (zvýšená vzdialenosť v smere x-ovej osi. Určuje sa s pravdepodobnosťou 50 percent.
- 5) Smer protihráča vzhľadom na pozíciu adresáta prihrávky (vľavo/vpravo).

Pri tréningu rozoznávame dva tímy. Prvý tím obsahuje hráčov, ktorých chceme trénovať. Druhý tím asistuje pri tréningu. Zber dát prebieha formou zápisu do súboru vo formáte XML. Je dôležité poznamenať, že zapisujú obidva tímy, s mierne pozmeneným formátom zápisu a v inom čase.

Hráč prvého tímu zapisuje v okamihu prihrávky spoluhráčovi. Hráč druhého tímu zapisuje v okamihu, keď získa loptu. Na základe uvedených zápisov je možné zisťovať úspešnosť alebo neúspešnosť danej prihrávky.

XML formát umožňuje vo svojej podstate vytvárať hierarchický zápis jednotlivých entít tak, aby údaje, ktoré k sebe logicky patria boli pokope, čo má za následok lepšiu prehľadnosť a dobrú čitateľnosť dokumentu.

Ďalej uvádzame formát XML súboru jednotlivých tímov spolu s vysvetlením a vyznačením údajov, ktoré budú slúžiť ako vstupy do neurónovej siete:

Tím č. 1:

```
<Prihravka super="0" typ="0"> //typ = 4. vstup
<Cas>0</Cas>
<Kto>
  <Cislo>2</Cislo>
  <Xcoord>-14.480124</Xcoord>
  <Ycoord>-13.591961</Ycoord>
</Kto>
<Komu>
  <Cislo>7</Cislo>
  <Xcoord>-5.562594</Xcoord>
  <Ycoord>14.303205</Ycoord>
</Komu>
<VektorKtoKomu>
  <Dlзка>29.285877</Dlзка> //1. vstup
</VektorKtoKomu>
<Protivnik>
  <Cislo>7</Cislo>
  <Xcoord>1.541600</Xcoord>
  <Ycoord>-23.907778</Ycoord>
  <VektorKtoEn>
    <Dlзка>19.055489</Dlзка> //2. vstup
    <KolmaV>20.000000</KolmaV> //3. vstup
    <Smer>1</Smer> //5. vstup
  </VektorKtoEn>
</Protivnik>
```

Vysvetlenie formátu

- Atribút **super** hovorí, o ktorý tím sa jedná. (0 = tím, ktorý trénuje, 1 pomocný tím)
- Atribút **typ** hovorí o tom, či sa jedná o prihrávku na telo (0) alebo do behu (1)
- Značka **Cas** predstavuje počet uplynutých cyklov hry
- Značka **Kto** predstavuje informácie o prihrávajúcom hráčovi, konkrétne číslo dresu, x a y súradnice
- Značka **Komu** predstavuje informácie o adresátovi prihrávky, konkrétne číslo dresu, x a y súradnice
- Značka **VektorKtoKomu** obsahuje informáciu o dĺžke vektora prihrávajúceho hráča a adresáta prihrávky
- Značka **Protivnik** združuje informácie týkajúce sa konkrétneho protivníka a jeho vzťahu k prihrávajúcemu hráčovi a adresátovi prihrávky. Značka sa vyskytuje za sebou toľkokrát, koľko súperov vidí prihrávajúci hráč.
- Značka **VektorKtoEn** obsahuje informáciu o veľkosti priemetu vektora prihrávajúceho hráča a protihráča do vektora prihrávajúceho hráča a adresáta prihrávky, vzdialenosť protihráča od vektora prihrávajúceho hráča a adresáta

prihrávky a smer protihráča vzhľadom na adresáta prihrávky(1 –bližšie k súperovej bráne, 0 – ďalej od súperovej brány)

Tím č. 2

```
<Prihravka super="1">
  <Cas>0</Cas>
  <Kto>
    <Cislo>2</Cislo>
    <Xcoord>-14.480124</Xcoord>
    <Ycoord>-13.591961</Ycoord>
  </Kto>
</Prihravka>
```

Vysvetlenie formátu

Formát tímu č. 2 vychádza z formátu prvého tímu. Jeho údaje slúžia len na zisťovanie neúspešnosti prihrávok a nevstupujú priamo do neurónovej siete.

Po skončení zápasu zozbierané údaje odchádzajú do transformátora, ktorý ich vyhodnotí a pretransformuje na vstupy do neurónovej siete. O transformátore píšeme v inej časti dokumentu.

Alternatívna reprezentácia

Pri zvažovaní vhodných vstupov do neurónovej siete sme sa rozhodli vyskúšať využitie viacerých údajov, s cieľom lepšieho opísania danej situácie a teda aj lepšieho naučenia siete. Rozšírili sme aktuálny formát XML súboru o rýchlosti hráčov a lopty a kosínusov uhlov.

Bohužiaľ, v procese tréningu zozbieraných vzoriek sa vyskytli anomálie, keď sa sieť dokázala vzorku naučiť neprirodzene rýchlo (niekoľko epoch). Samozrejme, výsledky zápasu potvrdili naše obavy a ukázali, že sa sieť natrénovala nesprávne.

Príčinu zlyhania zatiaľ nevieme presne stanoviť, ale domnievame sa, že môže súvisieť s veľkým množstvom podobných vstupov na rôzne ohodnotenia.

Kým sa nám nepodarí zistiť a odstrániť všetky súvisiace problémy, budeme naďalej používať prvú alternatívu výberu a vyhodnocovania herných údajov .

6.5.3 Spracovanie tréningových dát

Na spracovanie dát z tréningu sa používa XSLT (Extensible Stylesheet Language Transformations). Spracovanie je rozdelené do dvoch fáz: Zlúčenie dokumentov všetkých hráčov do jedného chronologicky usporiadaného súboru vo formáte XML. Tento súbor sa nazýva `hra.txt`.

Z pozorovania sme zistili, že v rovnaký čas zapisujú viacerí hráči. Takéto údaje sme vylúčili, keďže môžu spôsobiť skreslenie. Do úvahy sa berie iba prvý záznam s rovnakým časom. Ďalšími situáciami, ktoré môžu skresliť výsledok je napr. aut. Súčasní hráči nereagujú dostatočne rýchlo na tieto situácie, preto dochádza k spomaleniu. Toto

pozorujeme na udalostiach tak, že rozdiel časov je veľký. Z pozorovania sme odhadli čas na 40 cyklov.

Prihrávku ohodnotíme ako úspešnú, ak nasledujúci záznam je od spoluhráča s rovnakým číslom ako číslo hráča, komu hráč prihrával. V tomto prípade spracujeme všetky pozície protihráčov, ktorých čísla poznáme, a získame tak niekoľko vstupov pre tréovanie neurónovej siete.

Prihrávka je ohodnotená ako neúspešná, ak nasledujúci záznam je od protihráča, ktorého číslo hráč poznal. V takomto prípade spracujeme len jeden záznam protivníka a to toho, ktorý zachytil prihrávku.

Všetky ostatné situácie sa vyhodnotia ako neidentifikované a ďalej sa nespracúvajú. Ide napr. o zachytenie lopty spoluhráčom, ktorému nebola lopta určená.

Výstupom je textový dokument, ktorý sa môže použiť pre tréovanie neurónovej siete. Zo zaznamenaných informácií je možné zistiť aj určité vlastnosti hry. Napríklad ako často sa jednotliví hráči dostanú do kontaktu s loptou.

Podpora matematických operácií nie je veľmi dobre podporovaná jazykom XSL, preto všetky výpočty prebiehajú pred samotným spracovaním. Výhodou XSL je ľahšia čitateľnosť, než pri vlastne vyrobených parseroch. Umožňuje rýchle zmeny napr. pri pridání nových parametrov alebo zmene poradia.

6.6 PRÍČINA PÁDU KOUČA

Počas letného semestra sa nám podarilo odhaliť chybu v koučovi, ktorá spôsobovala jeho „pád“ (t.j. predčasné ukončenie). Kouč využíva súbor `units.xml`, nesúci informácie o formáciách. V tomto súbore boli na niekoľkých miestach zle popárované otváracie a uzatváracie XML značky. Túto chybu sme odstránili a potom kouč fungoval bez problémov.

6.7 NAJDÔLEŽITEJŠIE ZMENENÉ A DOPLNENÉ FUNKCIE

V tejto časti budú popísané funkcie nami vytvoreného prototypu. Niektoré funkcie pôvodného hráča FC Farmári sme nechali s pôvodnými menami, ale ich obsah sa zmenil.

```
void FM_DribbleTo(const Point& pos)
```

Funkcia sa nachádza v triede `PlayerSkills`. Funkcia rozdelí priestor okolo spojnice k cieľu (súradnice cieľa sú v bode `pos`) na body, ktoré sú vzdialené o od hráča o rovnakú vzdialenosť (akýsi polomer), ale každý z nich je od osi spojnice otočený pod iným uhlom. Uhly od spojnice k cieľu sú nasledovné -80° , -70° , -60° , -50° , -40° , -30° , -20° , -10° , 0° , 10° , 20° , 30° , 40° , 50° , 60° , 70° , 80° . Z týchto bodov sa vyberie ten, ktorý je najmenej ohrozený spomedzi všetkých protihráčov. Následne sa táto hodnota porovná s polomerom a v prípade, že je väčšia tak, hráč kopne loptu do tohto bodu a nastaví premennú `FM_TIME` v závislosti od polomeru. Ak je menšia polomer sa zníži a začne nové vyhodnocovanie.

```
float FM_AnglePenalty(float x, Vector where, Vector enspd)
```

Funkcia sa nachádza v triede `PlayerSkills`. Funkcia slúži ako „penalizátor“ uhlov vo funkcii `FM_DribbleTo`. Vracia číslo väčšie ako 1, ktoré je tým väčšie čím je väčšia hodnota uhlu (premenná `x`). Vektory `where` (smer vektora kde sa ide kopat') a `enspd` (vektor rýchlosti protihráča) slúžia ako faktor vo výpočte, ktorý zohľadňuje ako veľmi musí súper zmeniť smer na to, aby zachytil loptu.

```
bool FCBelongsToArea(Point A, Point B, Point E)
```

Funkcia sa nachádza v triede `PlayerSkills`. Táto funkcia vracia boolovskú hodnotu reprezentujúcu to, či protihráč (v bode `E`) ohrozuje spoluhráča (v bode `B`). Pozícia strieľajúceho hráča je v bode `A`.

```
void FM_PassBallToPlayer(const Player& player, float spd, bool run)
```

Funkcia slúži na prihrávku spoluhráčovi. Na základe boolovskej premennej `run`, prihrá hráčovi `Player` na miesto (`run=false`) alebo do behu (`run=true`)

```
void PlayerSkills::FCGetDistances(Point A, Point B, Point E, int *sgnum, double* result)
```

Funkcia sa nachádza v triede `PlayerSkills`. Funkcia naplní pole `result` a premennú `sgnum` hodnotami pre neurónovú sieť.

```
void BehaveWithoutBall()
```

Funkcia sa nachádza v triede `PlayerTactics`. Funkcia slúži na správanie sa hráča bez lopty. Zmena nastala v tom, že pokiaľ má hráč nastavenú hodnotu `FM_TIME` väčšiu ako 0, je nútený bez ohľadu na iné správanie bežať za loptou.

```
void FM_DribbleNear(const Point& pos)
```

Funkcia, ktorá slúži ako dribbling na blízke vzdialenosti. Funkcia pracuje podobne ako `FM_DribbleTo`, ale používa väčší rozsah uhlov -170° do 170° s delením po 10° . Ďalší

rozdiel je, že v tomto prípade neexistuje žiadna penalizácia uhlov, takže lopta sa kope do bodu s najmenším ohrozením súpera.

Keďže pri testovaní sme zistili, že hráč používa často funkcie driblovania, ktoré sú pôvodné a nie naše bol kód upravený tak, aby sa pôvodné funkcie používali tesne pred súperovou bránou. Toto bolo dosiahnuté pomocou premennej `FM_CONST` a následným upravením, funkcií `SmartDribble` a `ShortKick`.

6.8 CHYBY V HRÁČOVI

Táto kapitola obsahuje stručný (a určite nie kompletný) zoznam nedostatkov a chýb hráča. Tieto chyby boli ešte súčasťou hráča FC Farmári, na ktorom sme postavili nášho hráča. Odhaľovali sme ich postupne, keď sme zistili, že hráč nerobí to, čo by sme od neho v danej situácii očakávali. Tento zoznam má poslúžiť ďalším tímom ako podklad pre stanovenie možných cieľov ich práce.

- 1) Na najnižšej úrovni hráča ide najmä o spracovanie lopty, t.j. jej zastavovanie, príp. zmenu jej smeru a rýchlosti. Hráč nie je schopný správne vypočítať smer a veľkosť vektora sily, ktorým má do pohybujúcej sa lopty kopnúť tak, aby sa lopta zastavila, alebo aby zmenila svoj smer na požadovaný. Tým má hráč sťažený príjem nahrávok. Túto funkcionalitu sa nám podarilo čiastočne vylepšiť, ale stále nefunguje ideálne.
- 2) Ďalej by bolo vhodné skontrolovať výpočtové vzorce použité pre prácu s vektormi na najnižšej úrovni.
- 3) Na úrovni rozhodovacej logiky je zlý algoritmus rozhodovania: Každý tím, ktorý dorábala nejakú novú hernú logiku hráča, umiestnil svoj kód na začiatok hráčovej rozhodovacej funkcie, čím došlo k nesprávnej voľbe priorit (napr. hráč sa najprv rozhoduje, či nahrá a až keď zistí, že nie je schopný úspešne nahráť, tak „uvažuje“ o strelbe na bránu. Priorita daných akcií by však mala byť presne opačná.) Bolo by vhodné sa zamyslieť nad vhodnosťou poradia výberu jednotlivých herných činností. Pri tom môže napomôcť vytvorená vizualizácia herného stromu hráča FC Farmári (viď. príloha H). Netreba však zabudnúť, že ide o vizualizáciu herného stromu pôvodného hráča FC Farmári, ktorá nezohľadňuje posledné zmeny, ktoré vykonal náš tím.
- 4) Správanie brankára: V zápase sa niekedy stáva, že brankár vybehne mimo šestnástky za loptou. Skôr než loptu dobehne, dostane sa k nej nejaký spoluhráč, ktorý ju odkopne kúsok ďalej od brány. V tom momente by sa mal brankár vrátiť na svoju domovskú pozíciu, on ale namiesto toho beží za loptou ďalej, pretože „si myslí“, že ju stále môže dobehnúť a odohrať. Takýmto spôsobom je nezriedka schopný dobehnúť takmer do polovice ihriska.
- 5) Hráč vidí väčší počet súperov ako je reálne možné z dôvodu nerozpoznania ich čísel.

7 ZHODNOTENIE

7.1 OVERENIE RIEŠENIA

Vzhľadom na špecifiká projektu nie je veľmi jednoduché povedať, či sme dosiahli zlepšenie riešenia. Overovanie prebiehalo subjektívnym pozorovaním hráčov pri hre. Sústreďovali sme sa na zmenené správanie a výsledky zápasu. Počas implementácie sme testovali takým spôsobom, že sme nastavili hráčovi len jednu práve testovanú funkcionálnu a spustili sme hru.

Posledné testy prebiehali s viacerými hráčmi z predchádzajúceho obdobia, pričom sme boli schopní so všetkými vyhrať, najmä s pôvodným hráčom FC Farmári. Z tohto pozorovania môžeme povedať, že sme dosiahli zlepšenie hráča.

7.2 ČO SME NESTIHLI

Plánovanú implementáciu libera a masiek sme nerobili vzhľadom na kvalitu pôvodného kódu, ktorý nám spôsoboval problémy nielen s implementáciou, ale aj riadením. Implementovaná funkcionálna je príliš prepojená, ťažko rozpoznať nejaké vrstvy alebo moduly, ktoré sú zodpovedné za danú funkcionálnu.

Z tohto dôvodu sme sa rozhodli úplne prepracovať existujúcu neurónovú sieť spolu s vytváraním tréningových údajov. Pracovali sme s novými parametrami, použili sme nové spôsoby rozhodovania a potýkali sme sa najmä s nepresným kovaním, ktoré je súčasťou základných funkcií. Napriek našej snahe o jeho opravu to nebolo v rámci časových možností uskutočniteľné. Bolo by potrebné prepracovať takmer kompletného hráča.

Použitie viacerých neurónových sietí sme neuskutočnili nielen z časových dôvodov, ale aj z dôvodu nekvalitnej pôvodnej siete. Preto sme sa viac sústredili na implementáciu jednej dobrej siete.

7.3 ČO SME SA NAUČILI

Počas týchto dvoch semestrov sme sa naučili, resp. zlepšili v tímovej práci. Počas implementácie sme boli často závislí pri nadväzujúcich úlohách, čo nás viedlo aj k zodpovednejšej práci.

Veľkým poučením pre nás bola dôležitosť dokumentácie a komentovaného zdrojového textu v tímových projektoch, čo bol nedostatok východiskového projektu.

Z praktickej skúsenosti sme získali ďalšie poznatky z neurónových sietí, XML a multiagentových systémov.

7.4 ZÁVER

V tomto projekte sa nám podarilo zlepšiť existujúceho hráča. Vzhľadom na stav hráča z hľadiska zdrojového kódu nie je vhodné pridávanie ďalších stratégií, naši nasledovníci by sa mali zamerať na prepracovanie kódu, zlepšenie modularity a podrobné zdokumentovanie existujúcej funkcionality. Súčasný kód obsahuje veľa dobrých myšlienok, ale nie všetky sa v skutočnosti používajú, resp. ich použitie je obmedzené.

Naše zlepšenia spočívajú hlavne v lepšom rozhodovaní použitím neurónovej siete a novým spôsobom predkopávania. Súčasťou riešenia sú aj reálne použité nástroje na tréningy spolu s dokumentáciou.

POUŽITÁ LITERATÚRA

- [FARM2005] BELLUŠ, J. ET AL.: RoboCup – nové stratégie. Tímový projekt (tím FC Farmári). Slovenská technická univerzita v Bratislave, Fakulta informatiky a informačných technológií.
<http://www2.dcs.elf.stuba.sk/TeamProject/2004/team03/>, 2005.
(6. 11. 2005).
- [FORMA2005] FORMA, S.R.O.: Elektronický lexikón slovenského jazyka.
<http://www.forma.sk/onlines/slex/index.asp>, 2005.
(16. 11. 2005).
- [LAST2004] KRIŽO, E. ET AL.: RoboCup – nové stratégie. Tímový projekt (tím SKLO). Slovenská technická univerzita v Bratislave, Fakulta informatiky a informačných technológií.
<http://www2.dcs.elf.stuba.sk/TeamProject/2003/team08/>, 2004.
(16. 11. 2005).
- [RoboCupSS] The Robocup Soccer Server manual – manuál k soccerserveru
<http://sserver.sourceforge.net/docs/manual.pdf>
(16. 11. 2005).
- [SKLO2004] DRAHOŠ, P. ET AL.: RoboCup – nové stratégie. Tímový projekt (tím L.A.S.T United). Slovenská technická univerzita v Bratislave, Fakulta informatiky a informačných technológií.
<http://www2.dcs.elf.stuba.sk/TeamProject/2003/team05/>, 2004.
(16. 11. 2005).
- [SQUIRREL2005] POZOR, M. ET AL.: RoboCup – nové stratégie. Tímový projekt (tím Squirrel Squadron). Slovenská technická univerzita v Bratislave, Fakulta informatiky a informačných technológií.
<http://www2.dcs.elf.stuba.sk/TeamProject/2004/team01/>, 2005.
(6. 11. 2005).
- [UvA Trilearn] Prof. dr. ir. F.C.A. Groen, dr. N. Vlassis, drs. J.R. Kok - University of Amsterdam
<http://staff.science.uva.nl/~jellekok/robocup/>
(16. 11. 2005).
- [FCPort2001] LAU N., REIS L. P.: FC Portugal 2001 Team Description: Flexible Teamwork and Configurable Strategy, 2001
<http://www.ieeta.pt/robocup/archive.htm>
(16. 11. 2005).