



Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Ilkovičova 3, 842 16 Bratislava 4



Tvorba softvérového systému v tíme
„RoboCup“ – nové stratégie
Analýza, špecifikácia, hrubý návrh

Tím č.6 – UTTP:

Bc. Peter Kajsa, Bc. Michal Kasan, Bc. Branislav Kontút,
Bc. Dávid Kováč, Bc. Martin Nepšinský, Bc. Tibor Sekereš

Študijný program: Softvérové inžinierstvo/Informačné systémy

Ročník: 1, Inžinierske štúdium

Email: team06@googlegroups.com

Web: <http://www2.dcs.elf.stuba.sk/~team06is-si/>

Vedúci projektu: Ing. Marián Lekavý

Školský. rok: 2007/2008

I. Dokumentácia k projektu

| | | |
|----------|--|-----------|
| 1 | ÚVOD | 3 |
| 1.1 | ZADANIE | 3 |
| 1.2 | CIELE | 4 |
| 1.3 | ŠTRUKTÚRA DOKUMENTU | 4 |
| 2 | ANALÝZA EXISTUJÚCICH HRÁČOV | 5 |
| 2.1 | 6TH SENCE | 5 |
| 2.2 | FIITBA | 6 |
| 2.3 | BRAINSTROMERS | 8 |
| 2.4 | LOPTOŠI | 11 |
| 2.4.1 | <i>Analýza kvality zdrojový súborov.....</i> | <i>14</i> |
| 2.4.2 | <i>Analýza správání hráča Loptošov.....</i> | <i>15</i> |
| 2.5 | SPRÁVANIE ROZOHRÁVAJÚCEHO HRÁČA | 18 |
| | <i>Rohový kop.....</i> | <i>18</i> |
| | <i>Voľný kop, Individuálny voľný kop.....</i> | <i>19</i> |
| | <i>Vhadzovanie.....</i> | <i>19</i> |
| | <i>Zahrávanie.....</i> | <i>19</i> |
| | <i>Pokutový kop.....</i> | <i>20</i> |
| | <i>Rozohranie.....</i> | <i>20</i> |
| 2.6 | SPRÁVANIE HRÁČA S LOPTOU POČAS HERNÉHO MÓDU | 21 |
| | <i>Strela na bránku.....</i> | <i>21</i> |
| | <i>Prekopnutie ofsajdovej línie.....</i> | <i>21</i> |
| | <i>Driblovanie.....</i> | <i>21</i> |
| | <i>Prihrávka1.....</i> | <i>22</i> |
| | <i>Prihrávka2.....</i> | <i>23</i> |
| | <i>Prihrávka3.....</i> | <i>23</i> |
| | <i>Predkopnutie lopty.....</i> | <i>23</i> |
| | <i>Strategická prihrávka.....</i> | <i>24</i> |
| | <i>Prihrávka na náhodnú pozíciu dáždnika.....</i> | <i>24</i> |
| 2.6.1 | <i>Analýza funkcií komunikujúcich so serverom</i> | <i>26</i> |
| 2.7 | ZHODNOTENIE ANALÝZY | 29 |
| 3 | ŠPECIFIKÁCIA | 30 |
| 4 | HRUBÝ NÁVRH | 31 |
| 4.1 | REFAKTORIZÁCIA A VYTIAHNUTIE SPRÁVANÍ | 31 |
| 4.2 | LOPTOŠOVSKÁ „KONCEPCIA VEDÚCEHO HRÝ“ A ZMENA SPRÁVANIA | 34 |
| 5 | PROTOTYP..... | 36 |
| 6 | ZÁVER | 37 |
| 7 | POUŽITÁ LITERATÚRA A ZDROJE | 38 |

1 ÚVOD

Tento dokument pojednáva o simulácii robotického futbalu v 2D priestore s názvom „RoboCup“. Dokument je projektovou dokumentáciou k predmetu „Tímový projekt“ v školskom roku roku 2007-2008. Dokument pozostáva z 2 základných celkov a to „Dokumentácia k projektu“ a „Riadenie projektu“. Dokument odráža proces tvorby softvérového systému. Sú tu opísané všetky fázy vývoja softvéru. Táto verzia odovzdávaná ku dňu 15.11.2007 obsahuje analýzu, špecifikáciu a hrubý návrh. Súčasťou tohto dokumentu je taktiež dokumentácia k riadeniu projektu.

V nasledujúcich fázach projektu sa bude tento dokument rozširovať o ďalšie časti projektovej dokumentácie.

1.1 Zadanie

Téme „RoboCup“, presnejšie lige simulovaného robotického futbalu sa naši študenti venujú už osem 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 dost' 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. Posledných niekoľko rokov sme preto ako podnázov použili "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 sú súčasťou celkového hodnotenia tohto projektu.

1.2 Ciele

Cieľom tohto projektu je refaktorovať jedného z hráčov fakulty, ktorí sa dlhodobo vyvíjajú na Fakulte informatiky a informačných technológií. Je pritom potrebné od základov prerobiť jeho štruktúru tak, aby boli jednotlivé časti hráča (jeho správania) samostatné, dali sa jednoducho meniť a nahrádzať a taktiež aby bolo možné jednoznačne identifikovať spoluprácu modulov správania. Táto refaktorizácia výrazne uľahčí štúdium a následné zlepšovanie hráča pre tímy v budúcich rokoch.

1.3 Štruktúra dokumentu

Tento dokument pozostáva z dvoch základných častí a to „Dokumentácia k projektu“ a „Riadenie projektu“.

„Dokumentácia k projektu“ popisuje etapy v procese tvorby softvérového produktu. Konkrétne sa jedná o výstupy každej etapy (analýzu, špecifikáciu požiadaviek a návrh systému). „Dokumentácia k projektu“ prezentuje naše výsledky, ktoré sme doposiaľ dosiahli.

Druhá časť dokumentu „Riadenie projektu“ má iný charakter. Táto časť zahŕňa dokumenty a iné doplňujúce informácie súvisiace s komunikáciou v tíme a riadením projektu. Sú tu zachytené spôsoby komunikácie, koordinácie členov tímu a samotné riadenie projektu.

Ďalšou neoddeliteľnou súčasťou druhej časti dokumentu sú prílohy, to jest dokumenty vytvárané v priebehu semestra.

2 ANALÝZA EXISTUJÚCICH HRÁČOV

Táto kapitola sa venuje analýze hráčov 4 tímov, pričom 3 tímy sú tímy fakultné (6th Sence, FIITBA, Loptoši), a potom nás zaujal jeden z najúspešnejších svetových tímov: Brainstormers. Analýza sa zameriava hlavne na architektúru hráča, čo vyplýva zo stanovených cieľov projektu. V jednotlivých analýzach sú uvedené najdôležitejšie časti, pričom sú vypichnuté výhody a nevýhody jednotlivých návrhov a riešení. Zaujímavou časťou analýzy je podkapitola zameraná na analýzu kvality zdrojový súborov jedného z tímov.

2.1 6th Sence

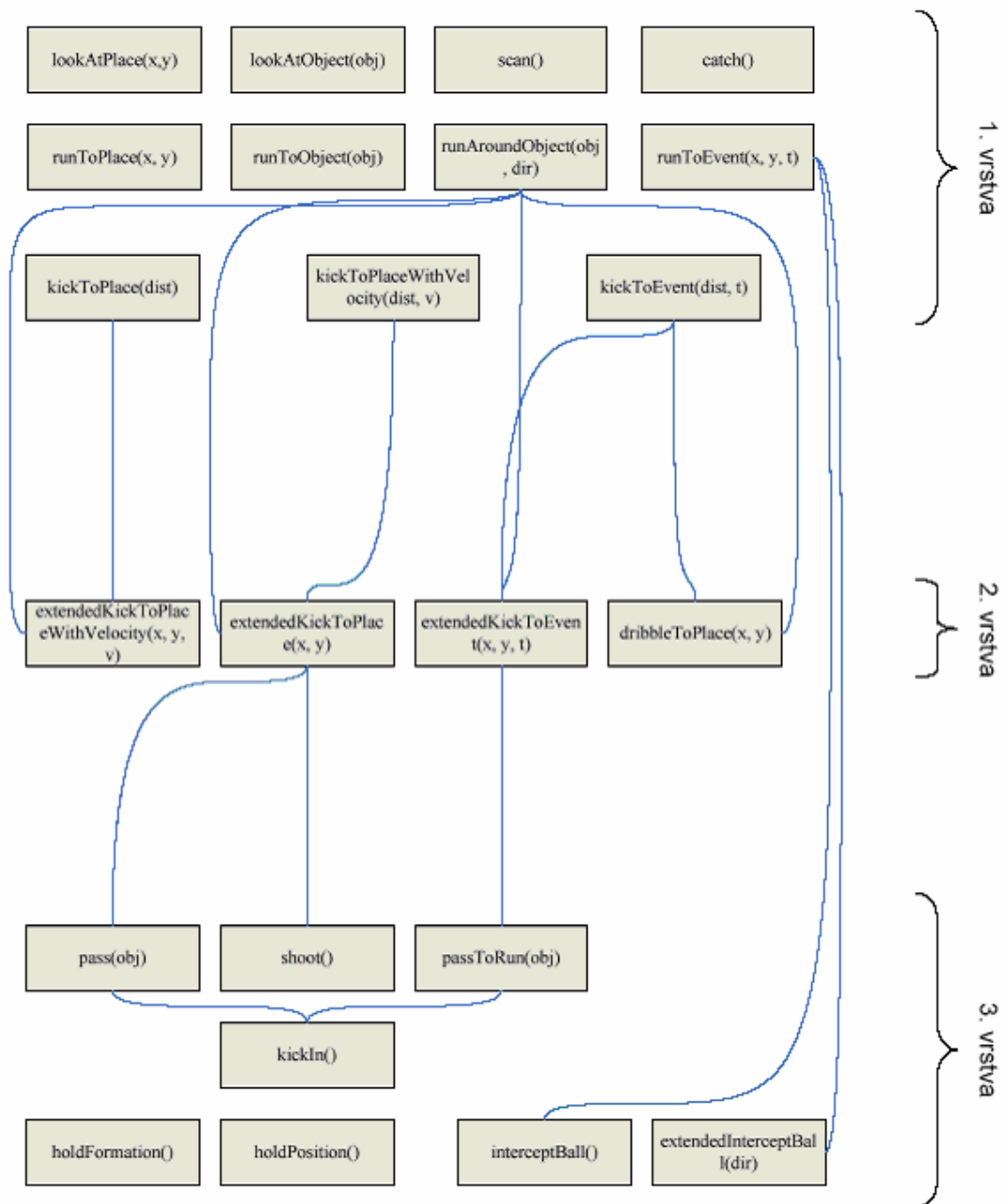
3D tím 6th Sence (školský rok 2006/07) si zobral pre svoj základ existujúceho hráča fakulty, konkrétne hráča tímu Hazard. Cieľom tímu 6th Sence bolo vytvorenie hráča s pokročilejšími schopnosťami a vyspelejším rozhodovaním. Svoju pozornosť upriamili predovšetkým na kvalitný návrh a implementáciu základných schopností hráča, ktoré mu umožňujú spracovávať vnemy prostredia, pohybovať sa po hracej ploche, prípadne narábať s loptou. Ďalej si dali za cieľ navrhnúť a implementovať modul vyššieho správania s rozhodovaním a plánovaním. Rozhodli sa, že jednoduché funkcionality hráča ako kopanie, behanie a hľadanie lopty matematickým výpočtom prihladnuc na fyzikálne zákony, a nie cez neurónové siete. Zmenili prístup k základným činnostiam hráča z neurónových sietí učených pomocou reinforcement learning-u na analytické, priamočiare rutiny.

Základné (nižšie) schopnosti hráča mali slúžiť vyšším schopnostiam, ktoré ich k svojej realizácii využívajú. Preto implementácia týchto nižších schopností mala byť prehľadná, modulárna, jednoducho rozširovateľná a kvalitne zdokumentovaná. Jedno správanie je v ich hrubom návrhu zapuzdrené práve v jednom module. Takéto rozdelenie je výhodné z hľadiska vyjadrenia závislostí. Konkrétnymi príkladmi vyššieho správania sú prihrávka (pass), prihrávka do behu (pass to run), strela na bránu (shoot). Hierarchicky vyššie od týchto správání je vykopnutie (kickIn). Príkladmi úrovňovo nižšieho správania sú KickToPlace(dist), KickToPlaceWithVelocity(dist, v), KickToEvent(dist, t), runToPlace(x, y), runToObject(obj) a iné.

Moduly architektúry hráča tímu 6th Sence, ich závislosti a taktiež úroveň abstrakcie modulov znázorňuje obrázok číslo 3. Vyžívanie funkcionality (správania) iných modulov považujeme za efektívny spôsob zabezpečenia vlastnej funkcionality. Môžeme hovoriť o spolupráci modulov. Tým, že je využitý kód modulu nižšej úrovne, je odstránená duplicita v kóde, keďže úrovňovo nižšie správanie je implementované len na jednom mieste, a je využívané viacerými modulmi. Takéto zapuzdrenie zvyšuje taktiež prehľadnosť kódu. Dôležitá je taktiež orientácia v zdrojovom kóde hlavne pri oboznamovaní sa so systémom. Táto orientácia v kóde bude pri takejto modularizácii určite lepšia. Vďaka tomu sa systém ľahšie udržiava a pružnejšie reaguje na prípadné požiadavky na zmenu. Jednoduché pridávanie nových funkcií agentovi je vlastnosť, ktorú si dal tento tím ako jeden z hlavných cieľov (rozširovateľnosť). Za najdôležitejšiu kvalitatívnu vlastnosť systému (nemá nič spoločné s funkcionalitou agenta) považujeme zameniteľnosť modulov (správání). Túto vlastnosť bude možné neskôr dokonale využiť pri testovaní jednotlivých správání. Takto bude možné nahrádzať jedno správanie za druhé a následným testovaním bude možné overiť, ktoré správanie je lepšie vzhľadom k dosiahnutiu cieľa. Aby sme mohli zamieňať moduly, je

Analýza existujúcich hráčov

nevyhnutná vzájomná nezávislosť modulov, ktorú je možné dosiahnuť použitím rozhraní. Nijaký nami skúmaný tím nemal správanie sústredené v jednom module, a tým sa architektúra tímu 6th Sense výrazne odlišila od ostatných.



Obr. 1 - Návrh hierarchie modulov správania

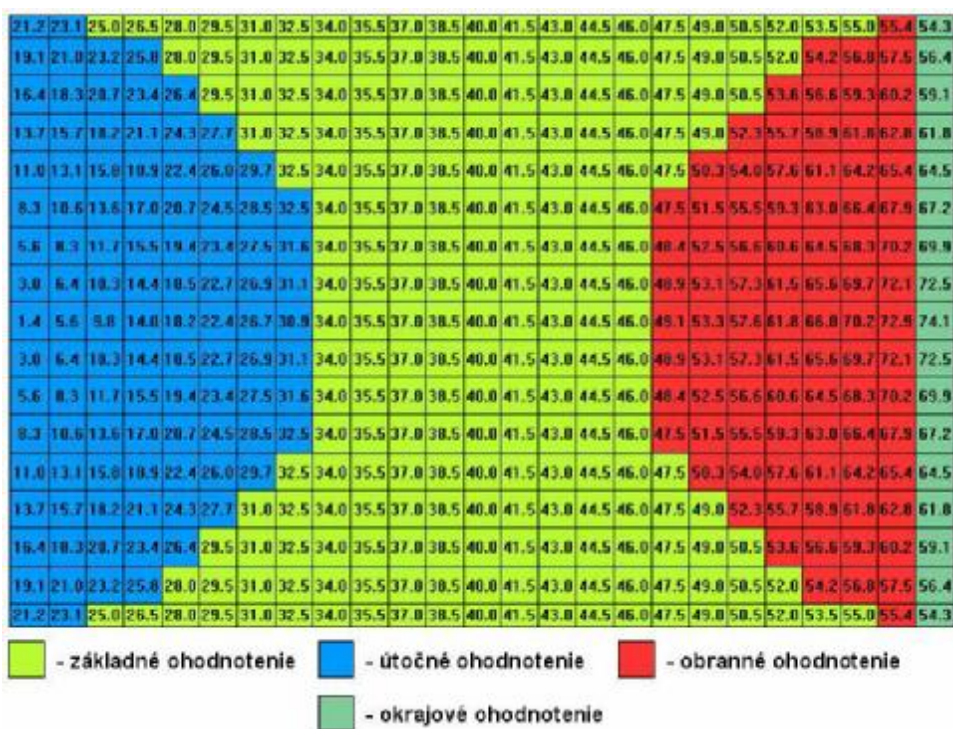
2.2 FIITBA

Z tímu FIITBA vychádza aj tím Loptoši, ktorý sme sa rozhodli modularizovať a spolu sa zúčastnili na turnaji „RoboCup“ at FIIT'07, na ktorom tím FIITBA obsadil druhé a Loptoši

Analýza existujúcich hráčov

tretie miesto. Oba tímy mali rovnaký počet bodov, ale tím FIITBA inkasoval o jeden gól menej, čo svedčí o podobnosti oboch tímov. Tím FIITBA, ktorý vznikol ako diplomový projekt, nadväzuje a snaží sa rozvíjať tím FC Farmári, ktorý vznikol o rok skôr. Štruktúra tímu FIITBA je podobná ako pri tíme Loptoši. Autor sa venoval aj refaktorizácii kódu, tak aby bol prehľadnejší a ľahšie modifikovateľný, ale nepodarilo sa mu to vykonať do akceptovateľného stavu, z čoho vyplýva aj naše zadanie. Následne sa autor venoval vylepšeniu rôznych vlastností hráčov.

Jeden z prínosov hráča FIITBA je implementovanie modulu vrstevnicovej mapy na rozhodovanie hráča a na reprezentáciu informácií o svete. Potreba tohto modulu vznikla z celkovej ignorácie spoluhráčov ale hlavne súperových hráčov, ktorá bolo viditeľná pri tíme FC Farmári. Celé ihrisko je rozdelené na políčka rovnakej veľkosti a každé políčko je inicializované na určitú počiatočnú hodnotu (Obr. 2). Hráč doplní ohodnotenie políčok, ktoré sa nachádzajú v jeho blízkosti podľa svojich vizuálnych vnemov. Následne sa bude snažiť dostať na políčko s vyšším ohodnotením ako je ohodnotenie jeho aktuálneho políčka.



Obr. 2 - Inicializácia vrstevnicovej mapy

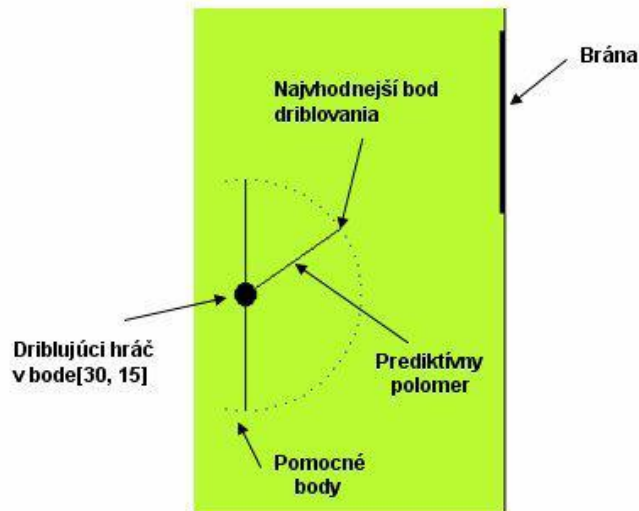
Podľa autora dosahoval hráč FIITBA použitím vrstevnicovej mapy lepšie výsledky ako pôvodný hráč.

Dalším zlepšením je rozohrávanie hráča zo stredu ihriska, pri ktorom podľa analýzy autora až pri 50 % prípadov získa loptu súper. Rozohrávajúci hráč si ohodnotí svojich spoluhráčov a následne prihrá tomu z najvyšším ohodnotením. Autor pri testovaní dosiahol až 98 percentnú úspešnosť.

Jedným z vylepšení, na ktoré sa využila vrstevnicová mapa je zlepšenie prihrávok medzi spoluhráčmi. Hráči sa nesnažia vždy posunúť dopredu, ako to bolo pri tíme FC Farmári, ale rozhodnú sa na základe vrstevnicovej mapy, čo vedie aj k prihrávkam dozadu. Dalším prvkom je prenesenie hry pomocou prihrávky na druhú stranu ihriska. Pri sledovaní simulácie je ale zrejme, že prihrávanie nie je ideálne, pretože hráč často prihrá súperovi alebo

do voľného priestoru, kde sa nik nenachádza. Z tímovým nahrávaním úzko súvisí aj nabiehanie si hráčov na prihrávky. Ak tím útočí, tak sa hráči snažia dostať do políčka s najvyšším ohodnotením pre prihrávku. Ohodnotenie ovplyvňuje napríklad prítomnosť súperovho hráča. To znamená, že voľní hráči sa budú snažiť uniknúť súperovi a dostať sa do voľného priestoru.

Dôležitou vlastnosťou hráča je aj jeho schopnosť driblovania. Autor sa snažil vylepšiť spôsob zvolenia si bodu, do ktorého bude hráč driblovať (Obr. 3). Autorovi sa podarilo dosiahnuť očakávané zlepšenie a podľa neho hráč dosahuje fantastické výsledky zlepšenia držania lopty.



Obr. 3 - Výber bodu na driblovanie

Autor sa snažil vylepšiť aj ďalšie časti hráča ako napríklad:

- obohranie súperovho hráča
- vylepšenie zachytávania lopty (InterceptBall)
- synchronizácia vizuálnej informácie s otáčaním krku
- rozhodovacia logika hráča s loptou a bez lopty
- zahrávanie štandardných situácií
- manažment energie hráčov
- štatistické vyhodnotenie hry koučom

Tím FIITBA je podľa nás celkom úspešný a autor priniesol a zrealizoval množstvo nápadov na zlepšenie hry.

2.3 Brainstromers

Brainstormers patria k najúspešnejším tímom v "RoboCup"e. Sú víťazmi posledných majstrovstiev sveta (2007). Hlavnou motiváciou pri vývoji je objavovanie možností umelej inteligencie a strojového učenia. Obzvlášť sa sústreďujú na metódu učenia odmenou a trestom (UOT), ktorá bola úspešne použitá pre rozličné časti rozhodovacích modulov.

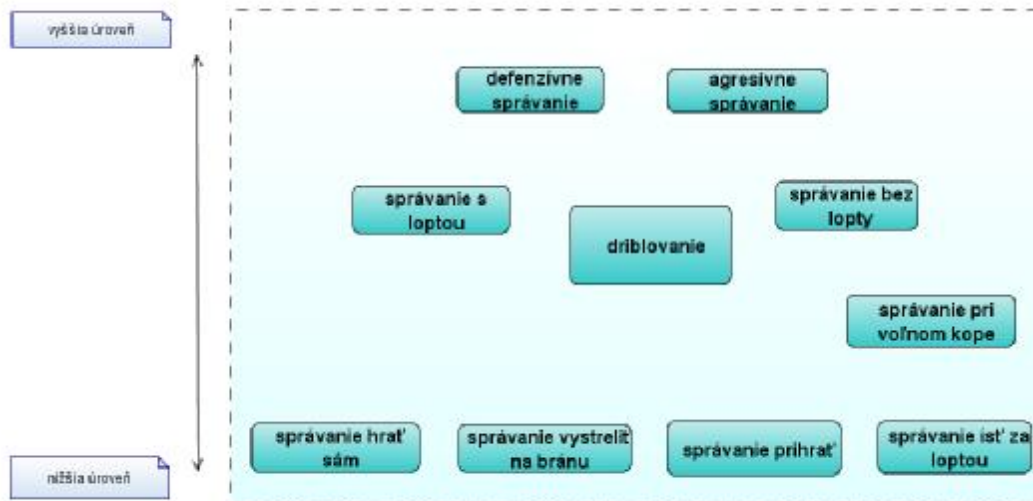
Brainstormers sa opierajú o nasledujúce základné princípy:

Analyza existujúcich hráčov

- sú dva základné moduly: modul sveta a rozhodovací modul
- vstupom do rozhodovacieho modulu je približný celkový model sveta
- prostredie futbalu je modelované ako markovovský rozhodovací proces
- rozhodovanie je organizované v komplexných a menej komplexných moduloch správania
- moduly správania sa učia pomocou UOT
- moderné metódy umelej inteligencie sú aplikované všade, kde to je len možné

Za úspechom tohto tímu stojí vylepšené učenie sa zručnosti hráča, učenie stratégií tímu UOT a vyváženie medzi reaktívnosťou a deliberatívnosťou.

Rozhodovací proces agenta je odvodený z behavior-based architektúry robota. Množina modulov správania realizuje rozhodovanie agenta. Táto architektúra je hierarchická, postupuje od komplexných správání smerom k špecializovaným. Nie je však čisto hierarchická, správanie z nižšej úrovne môže volať správanie z vyššej a na základe toho sa rozhodnúť o akcii.

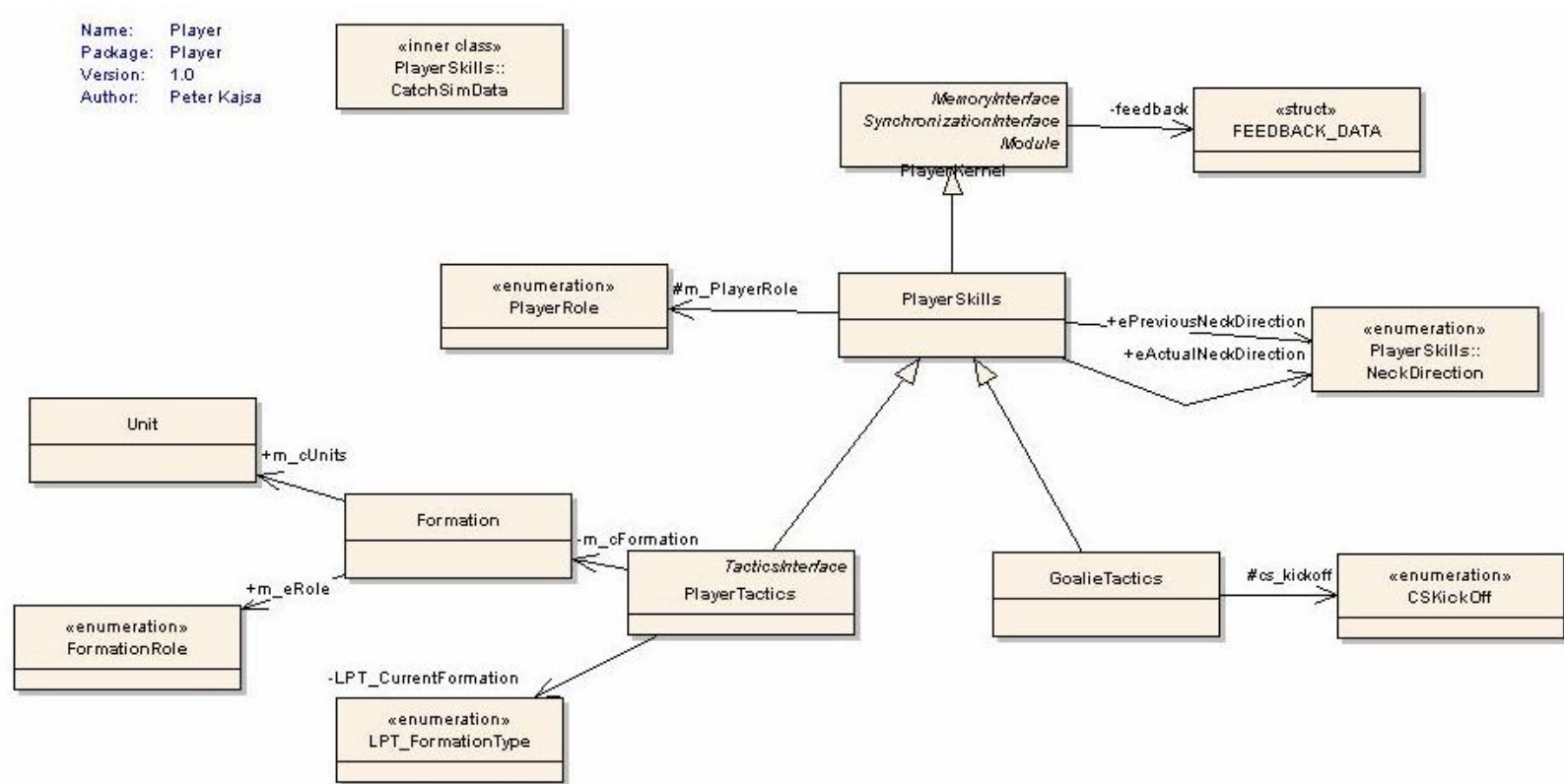


Obr. 4 - Architektúra správania

2.4 Loptoši

Loptoši vo svojom tímovom projekte postavili svojho hráča na základe predchádzajúceho projektu FIITMEDIA a neskôr FIITBA. Finálny hráč Loptošov teda vychádza z hráča tímu FIITBA. Z dokumentácie tímu Loptoši by sa mohlo zdať zrejme, že došlo k mnohým vylepšeniam hráča z hľadiska jeho architektúry resp. modulárnosti. Ako sami uvádzajú ich úprava kódu spočívala v implementovaní návrhových vzorov, ktoré zlepšili čitateľnosť kódu, odstránení nepoužitých abstraktných tried SkillsInterface a AbilitiesInterface, rozdelení dlhých funkcií na jednoduchšie a odstránenie zbytočného dedenia triedy Main od triedy Module, vyňatie kódu do novej funkcie pri duplicite kódu. Napriek uvedeným vylepšeniam, ktoré vysoko pravdepodobne neboli implementované v plnej miere, však nepovažujeme architektúru hráča Loptoši za veľmi úspešnú predovšetkým z hľadiska princípov dobrého objektovo orientovaného návrhu. Ako je zrejme z diagramu tried na obrázku 1, prípadne obrázku 2, v ktorom sú na rozdiel od obrázku 1 zachytené všetky objektové asociácie, medzi mnohými časťami hráča je použitá asociácia dedenia, ktorá predstavuje najsilnejšiu väzbu z hľadiska OOP a navyše jednotlivé triedy PlayerKernel, PlayerSkills, PlayerTactics a GoalieTactics sú veľmi rozsiahle, čo navyše spôsobuje s použitím asociácie dedenia nabaľovanie funkcionality. I samotná rozsiahlosť uvedených tried, ako je možné si všimnúť v diagrame tried dostupnom na adrese http://www2.dcs.elf.stuba.sk//team06isi/Loptosi_class_diagram_with_all.jpg, prispieva k tomu, že zdrojový kód a jednotlivá funkcionality sa stáva predovšetkým ťažko identifikovateľná a v neposlednej miere ťažko vymeniteľná. Za istý nedostatok architektonického riešenia hráča Loptoši taktiež považujeme zriedkavé používanie rozhraní resp. abstraktných tried, ktoré robia väzbu medzi objektmi voľnejšiu, keďže sa jednotlivé odkazy viažu na typ rozhrania. Vzhľadom na uvedené skutočnosti sme sa rozhodli pre refaktORIZÁCIU práve hráča Loptošov, pričom by sme chceli priniesť výrazné zlepšenie z hľadiska architektúry a modulárnosti hráča, keďže hráči „RoboCup“u majú pred sebou ešte dlhú evolúciu a je teda potrebné zabezpečiť, aby zmena určitej funkcionality už existujúceho hráča nebola utópiou v podobe analyzovania a menenia tisícok riadkov kódu.

Analyza existujúcich hráčov



Obr. 6 - Filtrovaný diagram tried hráča Loptoši

2.4.1 Analýza kvality zdrojový súborov

V tejto kapitole je pomocou rôznych nástrojov analyzovaný zdrojový kód tímu Loptoši z hľadiska jeho kvality. Existujú rôzne pohľady na kvalitu, pretože nie všetky konvencie sú presne určené. Medzi vlastnosti kvalitného kódu by mala patriť dobrá čitateľnosť, zrozumiteľnosť a ľahká modifikovateľnosť. Kvalita sa zvyšuje napríklad dodržiavaním konvencií daného programovacieho jazyka alebo nízkou cyklomatickou komplexnosťou, čo je počet rôznych ciest od začiatku do konca funkcie. Pri tomto projekte by podľa nás mal byť kladený zvýšený dôraz na kvalitu, pretože nové tímy väčšinou preberú prácu starších tímov a zmenia určité časti hráča. Značnú časť času musia venovať analýze súčasného stavu hráča a identifikovaniu správania hráča v kóde, ktoré chcú zmeniť. Pri tomto procese je okrem vyššie spomenutých vlastností dôležité aj správne okomentovanie zdrojového kódu. Pomocou vyššej kvality je možné aj predísť určitému množstvu chýb v kóde a zjednodušiť sa testovanie hráča. Na analýzu kvality sme použili dva nástroje: Resource Standard Metrics a Understand for C++.

Pomocou programu Understand for C++ sme analyzovali všetkých 136 zdrojových a hlavičkových súborov (Tab. 1). Program generuje vyhodnotenie vo forme HTML dokumentov. Vo vyhodnotení je analyzovaná každá trieda a funkcia a v prehľade sú o nej zobrazené rôzne informácie ako napríklad rôzne typy komplexnosti a počty rôznych typov riadkov (kód, komentár, prázdne riadky, ...). Pri každej funkcii a súbore je vypočítaný aj pomer komentáru k riadkom kódu. Pre celý projekt je tento pomer okolo 0,35. Zaujímavé je tiež zistenie o nepoužívaných triedach, funkciách a objektoch, ktoré je dostupné vo výslednom hodnotení.

| | |
|----------------------|-------|
| Triedy | 84 |
| Súbory | 136 |
| Funkcie | 1081 |
| Riadky | 35581 |
| Prázdne riadky | 5830 |
| riadky kódu | 21001 |
| riadky komentáru | 7282 |
| neaktívne riadky | 1410 |
| Deklaratívne príkazy | 5158 |
| Vykonaateľné príkazy | 12422 |
| pomer komentár/kód | 0,35 |

Tab. 1 - Zhrnutie analýzy programom Understand for C++

Program Resource Standard Metrics analyzuje zdrojové kódy a umožňuje zobraziť výstup vo formáte HTML. Poskytuje množstvo informácií o jednotlivých funkciách, triedach a súboroch a medzi inými aj rôzne nedostatky v kvalite zdrojového kódu. V Tab. 2 sú zobrazené niektoré z pripomienok programu Resource Standard Metrics ku kvalite zdrojového kódu. Pripomienky sú rôzneho druhu od nedodržania programovacích konvencií C++ až po nedostatky v správe pamäti.

| Pripomienka | počet výskytov |
|--|----------------|
| počet znakov v riadku > 80 | 1499 |
| if, else, for alebo while nie je ohraničené blokom | 1209 |
| chýba komentár pred definíciou funkcie | 416 |
| počet návratových bodov funkcie > 1 | 262 |
| funkcia obsahuje menej ako 10% komentáru | 189 |
| funkcia neobsahuje žiaden logický riadok kódu | 115 |
| špecifikácia triedy obsahuje ukazovateľ na dáta | 97 |
| špecifikácia triedy obsahuje protected dáta | 86 |
| cyklomatická komplexnosť > 10 | 85 |
| špecifikácia triedy obsahuje public dáta | 83 |
| ukazovateľ na ukazovateľ | 40 |
| ... ako parameter funkcie | 38 |
| pre inkrementačný ++ operátor | 37 |
| použitý ternárny operátor ? | 34 |
| konštanta definovaná pomocou #define | 24 |
| rôzny počet operátorov new a delete v súbore | 22 |
| chýba komentár pred definíciou triedy | 18 |
| blok switch neobsahuje možnosť default | 20 |
| priradenie v rámci deklarácie if alebo while | 12 |
| súbor obsahuje menej ako 10% komentáru | 11 |
| použitie ;; | 10 |
| Spolu: | 4307 |

Tab. 2 – Zhrnutie analýzy programom Resource Standard Metrics

Z analýzy kvality zdrojového kódu vyplýva, že kód obsahuje značné množstvo problémových miest. Väčšina z nich pramení z nedodržania programovacích konvencií jazyka C++, ale sú tam aj vážnejšie nedostatky, ktoré môžu ohroziť funkčnosť programu.

2.4.2 Analýza správání hráča Loptošov

Elementárne správania

Za elementárne správania považujeme správania, ktoré sa už ďalej nezjemňujú, ale priamo vyžadujú zaslanie správ na server.

Zmena pohľadu

Algoritmus:

Zmena pohľadu hráča sa uskutočňuje v metóde `PlayerSkills::NeckSynchronization()` pomocou metódy `CommandsInterface::ChangeView()`.

Závislosti:

objekty: `world` (získovanie informácií o pozícií, o čísle atď.)

funkcie: `CommandsInterface::ChangeView()`
`PlayerSkills::TurnNeckToAux()`

Lokalizácia:

Správanie Zmena pohľadu je lokalizované v dokumentácii tímu StjupidDox, ktorý toto správanie navrhol avšak neimplementoval.

Otočenie krku

Algoritmus:

Zmena natočenia krku sa uskutočňuje v metóde `PlayerSkills::TurnNeckToAux()` pomocou metódy `CommandsInterface::TurnNeck()`.

Závislosti:

objekty: `world` (získovanie informácií o pozícií, o číse atď.)

funkcie: `CommandsInterface::TurnNeck()`

Lokalizácia:

Správanie Otočenie krku je lokalizované v dokumentácii tímu StjupidDox, ktorý toto správanie ako prvý implementoval a tímu FIITBA, ktorý toto správanie zmenili na otáčanie krku.

Otočenie

Algoritmus:

Natočenie hráča sa uskutočňuje v metóde `PlayerSkills::TurnTo()` pomocou metódy `CommandsInterface::Turn()`.

Závislosti:

objekty: `world` (získovanie informácií o pozícií, o číse atď.)

funkcie: `CommandsInterface::Turn()`

Lokalizácia:

Správanie Otočenie patrí k elementárnym správaniam , ktoré poskytuje server.

Kopnutie

Algoritmus:

Kopnutie do lopty sa uskutočňuje v metóde `PlayerSkills::DumbKick()` pomocou metódy `CommandsInterface::Kick()`.

Závislosti:

objekty: `world` (získovanie informácií o pozícií, o číse atď.)

funkcie: `CommandsInterface::Kick()`
`PlayerSkills::TurnTo()`

Lokalizácia:

Správanie Kopnutie je lokalizované v dokumentácii tímu StjupidDox, neskôr sa pridalo kopnutie na bránu. Loptoši si však spravili vlastné správanie „Strela na bránu“.

Blízke kopnutie

Algoritmus:

Blízke kopnutie lopty sa uskutočňuje v metóde `PlayerSkill::NearKick()`. Najprv sa zistí, či je daný hráč dostatočne blízko k lopte. Potom zistíme potrebnú veľkosť rýchlosti kopnutia alebo či je potrebné najprv loptu zastaviť. Nakoniec samotné kopnutie do lopty vykonáme pomocou metódy `CommandsInterface::Kick()`.

Závislosti:

objekty: `world` (získovanie informácií o pozícií, o číslach atď.)

funkcie: `CommandsInterface::Kick()`

Lokalizácia:

Správanie Blízke kopnutie je lokalizované v dokumentácii tímu Roztoče, s týmto správaním úzko súvisí správanie dobehnutie lopty – `NearKick`.

Spoločné správania

Za spoločné správania považujeme tie správania, ktoré hráči využívajú v oboch stavoch, s loptou, aj bez lopty.

Spoznávanie okolia

Algoritmus:

Závislosti:

objekty: `world` (získovanie informácií o pozícií, o číslach atď.)

funkcie: `PlayerSkills::DumbKick()`

Lokalizácia:

Správanie Spoznávanie okolia je lokalizované v dokumentácii tímu Roztoče, avšak chýba popis algoritmu.

Správanie hráča s loptou

Správanie s loptou sa rozdeľuje na základe toho, či daný hráč sa nachádza v hernom móde alebo je určený na rozohrávanie. Toto rozdelenie závisí od návratovej hodnoty metódy `PlayerTactics::ShouldKickIn()` pre každého hráča. Hráč určený na rozohrávanie sa vyberie na základe najbližšej domovskej pozícií hráča k miestu rozohrávania. Hráč určený na rozohrávanie realizuje inštrukcie v bloku za podmienkou `if(ShouldKickIn())`, ostatní hráči realizujú inštrukcie bloku za `else`.

Prihrávanie

Algoritmus:

Prihrávanie sa vykonáva v metóde `PlayerSkills::PassBallTo()`. Najprv sa upraví rýchlosť prihrávky a potom sa vypočíta cieľová pozícia prihrávky. Prihranie sa vykoná v metóde `PlayerSkills::NearKick()` (identifikované ako správanie blízke kopnutie). Po prihrávke hráč natočí hlavu na cieľ prihrávky pomocou metódy `PlayerSkills::TurnNeckTo()` (identifikované ako správanie pozeranie)

Závislosti:

Analýza existujúcich hráčov

objekty: `world` (získovanie informácií o pozícií, o čísle atď.)

funkcie: `PlayerSkills::NearKick()`
`PlayerSkills::TurnNeckTo()`

Lokalizácia:

Správanie tímové nahrávanie je lokalizované v dokumentácii tímu FIITBA.

Pozeranie

Algoritmus:

Toto správanie je implementované v metóde `PlayerSkills::TurnNeckTo()`. Hráč sa pozrie na danú pozíciu pomocou metódy `PlayerSkills::NeckSynchronization()` (identifikované ako zmena pohľadu) iba ak je vypnutá synchronizácia krku s pohľadom.

Závislosti:

objekty: `world` (získovanie informácií o pozícií, o čísle atď.)

funkcie: `PlayerSkills::NeckSynchronization()`

Lokalizácia: Správanie prihrávky do behu je lokalizované v dokumentácii tímu Roztoče. Je možné nájsť pod názvom „sledovanie lopty(stratégia sledovania ihriska)“.

Správanie rozohrávajúceho hráča

Hra je pozastavená a čaká sa na našu rozohrávku (resp. vhadzovanie). Hráč, ktorý je určený na rozohrávku sa poobzerá okolo seba (metóda `PlayerTactics::ExamineSurroundings()`), a tím získa informácie z okolitého sveta. Toto správanie je identifikované ako spoznávanie okolia. Po spoznaní okolia nasleduje vykonanie inštrukcií prislúchajúcich danému stavu sveta. Ak sa nachádzame v hracom móde `PM_CornerKick_Our`, realizuje sa správanie identifikované ako rohový kop. Ak sa nachádzame v hracom móde `PM_FreeKick_Our` alebo `PM_IndFreeKick_Our`, realizuje sa správanie identifikované ako voľný kop, individuálny voľný kop. Ak sa nachádzame v hracom móde `PM_KickIn_Our`, realizuje sa správanie identifikované ako vhadzovanie. Ak sa nachádzame v hracom móde `PM_KickOff_Our` alebo `PM_AfterGoal_Their`, realizuje sa správanie identifikované ako zahrávanie. Ak sa nachádzame v hracom móde `PM_PenaltyKick_Our`, realizuje sa správanie identifikované ako pokutový kop.

Rohový kop

Algoritmus:

Hráč, ktorý ide uskutočniť rohový kop sa v prvom cykle natočí na bod 25,0 (stred súperovej polovice) pomocou metódy `PlayerSkills::TurnTo()` (identifikované ako správanie otočenie). Ďalších 29 cyklov hráč čaká na umiestnenie spoluhráčov. Nasleduje rozohranie spoluhráčovi (identifikované ako správanie rozohranie) pomocou metódy `PlayerTactics::StartPlayOnByPassToOurPlayer()`, ktorá má návratovú hodnotu `true` ak sa rozohralo spoluhráčovi, inak `false`.

Závislosti:

objekty: `world` (získovanie informácií o pozícií, o čísle atď.)

funkcie: `PlayerSkills::TurnTo()`
`PlayerTactics::StartPlayOnByPassToOurPlayer()`

Lokalizácia:

Správanie rozohrávanie štandardných situácií a zahrávanie zo stredy je lokalizované v dokumentácii tímu FIITBA. Na toto nadviazal tím Loptoši a vytvorili správanie rohový kop, aut, ktoré je teda lokalizované v dokumentácii tímu Loptoši.

Voľný kop, Individuálny voľný kop

Algoritmus:

Hráč, ktorý ide uskutočniť voľný alebo individuálny voľný kop sa v prvom cykle natočí na bod 52.5,0 (stred súperovej bránky) pomocou metódy `PlayerSkills::TurnTo()` (identifikované ako správanie otočenie). Ďalších 29 cyklov hráč čaká na umiestnenie spoluhráčov. Nasleduje rozohranie spoluhráčovi (identifikované ako správanie rozohranie) pomocou metódy `PlayerTactics::StartPlayOnByPassToOurPlayer()`, ktorá má návratovú hodnotu `true` ak sa rozohralo spoluhráčovi, inak `false`.

Závislosti:

objekty: `world` (získovanie informácií o pozícií, o číslach atď.)

funkcie: `PlayerSkills::TurnTo()`

`PlayerTactics::StartPlayOnByPassToOurPlayer()`

Lokalizácia:

Správanie voľný kop je lokalizované v dokumentácii tímu FIITBA.

Vhadzovanie

Algoritmus:

Hráč, ktorý ide uskutočniť vhadzovanie si najprv v prvom cykle zistí, či sa nachádza v hornej alebo dolnej časti ihriska. Ak sa nachádza v hornej časti ihriska, natočí sa na $x, y+15$ (kde x a y sú súradnice hráča, ktorý ide vhadzovať), ak sa nachádza v dolnej časti ihriska, tak sa hráč natočí na $x, y-15$. Natočenie sa uskutoční pomocou metódy `PlayerSkills::TurnTo()` (identifikované ako správanie otočenie). Ďalších 29 cyklov hráč čaká na umiestnenie spoluhráčov. Nasleduje rozohranie spoluhráčovi (identifikované ako správanie rozohranie) pomocou metódy `PlayerTactics::StartPlayOnByPassToOurPlayer()`, ktorá má návratovú hodnotu `true` ak sa rozohralo spoluhráčovi, inak `false`.

Závislosti:

objekty: `world` (získovanie informácií o pozícií, o číslach atď.)

funkcie: `PlayerSkills::TurnTo()`

`PlayerTactics::StartPlayOnByPassToOurPlayer()`

Lokalizácia:

Správanie rozohrávanie štandardných situácií a zahrávanie zo stredy je lokalizované v dokumentácii tímu FIITBA. Na toto nadviazal tím Loptoši a vytvorili správanie rohový kop, aut, ktoré je teda lokalizované v dokumentácii tímu Loptoši.

Zahrávanie

Algoritmus:

Analýza existujúcich hráčov

Hráč, ktorý je určený na zahrávanie najprv pomocou metódy `CEvaluatingPlayerList::GetKickOffPlayer()` získa hráča, ktorý je vhodný pre prihrávku. Ak takýto hráč existuje, tak zahrávajúci hráč mu zahrá loptu silou 1, inak zahrá loptu na -25,0 (stred obranného pásma) silou 0.7. Samotné zahrávanie (identifikované ako správanie prihrávanie) sa uskutoční pomocou metódy `PlayerSkills::PassBallTo()`.

Závislosti:

objekty: `world` (získovanie informácií o pozícií, o číse atď.)
`CEvaluatingPlayer` (hráč z vizuálnej informácie iného hráča)
`CEvaluatingPlayerList` (zoznam hráčov z vizuálnej informácie hráča)
funkcie: `PlayerSkills::TurnTo()`
`PlayerSkills::PassBallTo()`

Lokalizácia:

Správanie rozohrávanie štandardných situácií a zahrávanie zo stredy je lokalizované v dokumentácii tímu FIITBA. Na toto nadviazal tím Loptoši a vytvorili správanie rohový kop, aut, ktoré je teda lokalizované v dokumentácii tímu Loptoši.

Pokutový kop

Algoritmus:

Pokutový kop sa uskutoční pomocou metódy `PlayerSkills::LPT_KickToGoal()` (identifikované ako správanie výstrel na bránu).

Závislosti:

objekty: `world` (získovanie informácií o pozícií, o číse atď.)
funkcie: `PlayerSkills::LPT_KickToGoal()`

Lokalizácia:

Správanie pokutový kop je lokalizované v dokumentácii tímu FIITBA.

Rozohranie

Algoritmus:

Rozohranie sa uskutočňuje v metóde `PlayerTactics::StartPlayOnByPassToOurPlayer()`. Na začiatku sa vypočíta vhodná pozícia v okolí hráča pre normálnu prihrávku pomocou metódy `CEvaluatingPlayerList::CalculateNormalPassPosition()`. Ak existuje takáto pozícia vypočítame potrebnú silu prihrávky a prihráme pomocou metódy `PlayerSkills::PassBallTo()` (identifikované ako správanie prihrávanie). Ak takáto pozícia neexistuje, prihráme pomocou metódy `PlayerSkills::PassBallTo()` (identifikované ako správanie prihrávanie) na pozíciu 40,0 (útočné pásmo).

Závislosti:

objekty: `world` (získovanie informácií o pozícií, o číse atď.)
`CEvaluatingPlayerList` (zoznam hráčov z vizuálnej informácie hráča)
`CEvaluatingPlayer` (hráč z vizuálnej informácie iného hráča)
`CEvaluatingPanemap` (vyhodnocovacia mapa)
funkcie: `CEvaluatingPlayerList::CalculateNormalPassPosition`
`PlayerSkills::PassBallTo()`

Správanie hráča s loptou počas herného módu

Tieto správania sa využívajú iba v prípade, ak má hráč loptu a zápas je v hernom móde, ktorý nezahŕňa rôzne štandardné situácie ako sú rôzne kopy na bránu a vhadzovanie.

Strela na bránku

Algoritmus:

Rozhodovania o strele zabezpečuje metóda `PlayerTactics::ShouldKickOnGoal()`. Ak má hráč loptu, prioritne zvažuje možnosť vystreliť na súperovu bránu. Hráč určite vystrelí na bránu ak je vzdialenosť lopty od brány menšia ako 17. Naopak, nepokúsi sa vystreliť, ak je vzdialenosť lopty od brány väčšia ako 25. Ak je aktuálna vzdialenosť medzi 17 a 25, tak hráč zvažuje na základe ďalších kritérií. Jedno z kritérií je informácia o súperovom brankárovi. Ak hráč nevidí brankára, tak nevystrelí na bránku okrem prípadu, kedy je bránka prázdna, čo sa zistí podľa bodu, na ktorý sa hráč pozerá. Ak je viditeľný aj súperov brankár, tak sa hráč snaží vypočítať, či je možné streliť gól, podľa pozície súperovho brankára a ostatných hráčov.

Vypočítanie bodu, do ktorého hráč vystrelí plnou silou zabezpečuje metóda `PlayerSkills::LPT_KickToGoal()` (identifikované ako správanie výstrel na bránu).

Závislosti:

objekty: `world` (získovanie informácií o pozícií, o číslach atď.)

funkcie: `PlayerSkills::LPT_KickToGoal()`
`PlayerTactics::ShouldKickOnGoal()`

Prekopnutie ofsajdovej línie

Algoritmus:

Prekopnutie zabezpečuje metóda `PlayerSkills::OffsideLineBreach()`, ktorá využíva základné správania prihrávanie a pozeranie. Hráč sa pokúsi prekopnúť ofsajdovú líniu ak je od nej vzdialený maximálne 4,75 a nenachádza sa príliš hlboko v súperovom obrannom pásme. Toto správanie môže realizovať jedine hráč s číslom 8 alebo viac, čiže iba ofenzívny hráč.

Závislosti:

objekty: `world` (získovanie informácií o pozícií, o číslach atď.)

`CEvaluatingPlayerList` (zoznam hráčov z vizuálnej informácie hráča)

`CEvaluatingPlayer` (hráč z vizuálnej informácie iného hráča)

`CEvaluatingPanelsMap` (vyhodnocovacia mapa)

funkcie: `PlayerSkills::OffsideLineBreach()`
`CEvaluatingPlayerList::CalculateOptimalDribblePointAux`
`PlayerSkills::PassBallTo()`
`PlayerSkills::TurnNeckTo()`

Lokalizácia:

Zohľadňovanie ofsajdov a prekopnutie ofsajdovej línie pridali Roztoče. Ďalší postup : Sttjupid dox zmenili, FC Farmári zlepšili, FIITBA zmenili.

Driblovanie

Algoritmus:

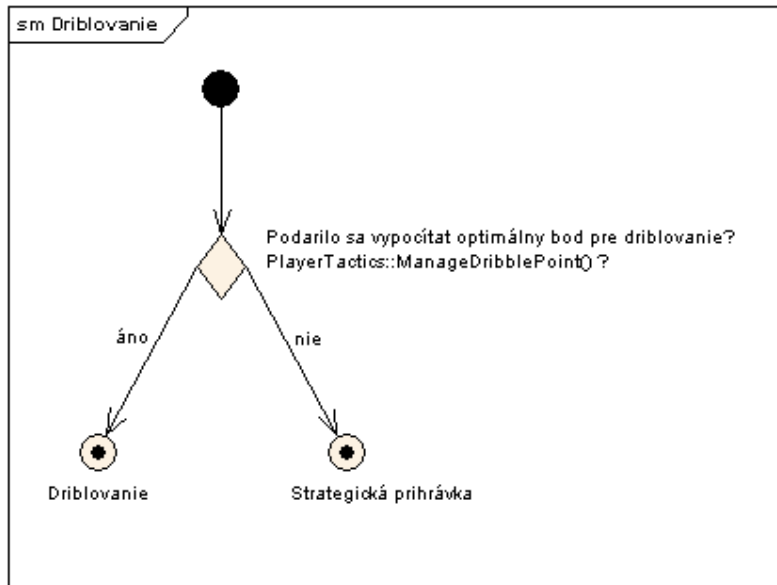
Rozhodnutie o bode driblovania zabezpečuje metóda `PlayerTactics::ManageDribblePoint()`, ktorá vypočíta súradnice bodu, do ktorého hráč bude driblovať a vráti TRUE alebo vráti

FALSE, ak nie je vhodné driblovať. Pri rozhodovaní hráča s loptou sa zvažuje driblovanie na troch miestach a ak nie je vhodné driblovať, tak sa volí správanie strategická prihrávka. Samotné driblovanie na vypočítaný bod vykonáva metóda `PlayerSkills::DribbleTo2()`. Na záver ak je hráč bližšie maximálne 35 metrov od súperovej brány, pokúsi sa pozrieť na súperovu bránu pomocou otočenia krku. Na (Obr. 8) je zobrazený stavový diagram pre driblovanie.

Závislosti:

objekty: `world` (získovanie informácií o pozícií, o čísle atď.)
`CEvaluatingPlayerList` (zoznam hráčov z vizuálnej informácie hráča)
`CEvaluatingPlayer` (hráč z vizuálnej informácie iného hráča)
`CEvaluatingPanelsMap` (vyhodnocovacia mapa)

funkcie: `PlayerSkills::DribbleTo2()`
`PlayerTactics::DistanceToGoal()`
`PlayerSkills::TurnNeckTo()`



Obr. 8 - Stavový diagram pre driblovanie

Prihrávka

Algoritmus:

Toto správanie sa využije ak sa podarí nájsť vhodnú pozíciu v okolí hráča pre prihrávku pomocou funkcie `CEvaluatingPlayerList::CalculateNormalPassPosition()` a nie je vhodné driblovať, pretože v okolí hráča sa nachádza súper (zistenie pomocou metódy `CEvaluatingPlayerList::NoEnemiesAround()`) alebo hráč je ďaleko od súperovej brány. Prihráva sa do bodu, ktorý vypočítala vyššie spomenutá metóda a sila sa určí podľa vzdialenosti. Prihranie sa uskutoční pomocou metódy `PlayerSkills::PassBallTo()` (identifikované ako správanie prihrávanie).

Závislosti:

objekty: `world` (získovanie informácií o pozícií, o čísle atď.)
`CEvaluatingPlayerList` (zoznam hráčov z vizuálnej informácie hráča)

Analýza existujúcich hráčov

CEvaluatingPlayer (hráč z vizuálnej informácie iného hráča)
CEvaluatingPanemap (vyhodnocovacia mapa)
funkcie: **CEvaluatingPlayerList::NoEnemiesAround()**
PlayerSkills::PassBallTo()

Prihrávka2

Algoritmus:

Ak sa nepodarilo nájsť vhodnú pozíciu v okolí hráča pre prihrávku pomocou funkcie **CEvaluatingPlayerList::CalculateNormalPassPosition()** a strieľať nie je vhodné, pretože brankár je veľmi blízko (bližšie ako 5), tak sa volí **Prihrávka2**, ktorej parametre sa určia pomocou pozície hráča a pozície brankára. Prihranie sa uskutoční pomocou metódy **PlayerSkills::PassBallTo()** (identifikované ako správanie prihrávanie).

Závislosti:

objekty: **world** (získovanie informácií o pozíciách, o číslach atď.)
CEvaluatingPlayerList (zoznam hráčov z vizuálnej informácie hráča)
CEvaluatingPlayer (hráč z vizuálnej informácie iného hráča)
CEvaluatingPanemap (vyhodnocovacia mapa)
funkcie: **PlayerSkills::PassBallTo()**

Prihrávka3

Algoritmus:

Ak neboli splnené podmienky pre vykonanie **Prihrávky2**, tak sa zvažuje použitie **Prihrávky3**. Toto správanie sa použije, ak je hráč hlboko v pásme súpera a v tomto pásme sú aj nejakí jeho spoluhráči. Zistenie spoluhráčov v pásme hráča sa urobí pomocou metódy **PlayerSkills::IsPointInRectArea()**. Prihranie sa uskutoční pomocou metódy **PlayerSkills::PassBallTo()** (identifikované ako správanie prihrávanie).

Závislosti:

objekty: **world** (získovanie informácií o pozíciách, o číslach atď.)
CEvaluatingPlayer (hráč z vizuálnej informácie iného hráča)
CEvaluatingPanemap (vyhodnocovacia mapa)
funkcie: **PlayerSkills::PassBallTo()**
PlayerSkills::IsPointInRectArea()
CEvaluatingPlayerList::OurPlayersInEnemyPenaltyArea()

Predkopnutie lopty

Algoritmus:

Vykonáva metóda **PlayerTactics::OutPlayOpponent()**. Hráč si predkopne loptu vľavo alebo vpravo od súpera. Toto je posledný variant, ktorý hráč vykoná vždy, ak nevykonal žiadnu inú akciu s loptou. Používa základné správania: prihrávanie, pozeranie.

Závislosti:

objekty: **world** (získovanie informácií o pozíciách, o číslach atď.)
funkcie: **PlayerTactics::GetNearestEnemy()**
PlayerSkills::PassBallTo()

Lokalizácia:

Predkopnutie lopty pridal tím FIITBA

Strategická prihrávka

Algoritmus:

Hráč sa pokúsi prihrať na strategickú pozíciu iba ak najbližší súper je vzdialený o viac ako 1,5. Smer tejto prihrávky nie je závislý od pozície žiadnych hráčov a ak je lopta ďaleko od brány, tak hráč kopne celou silou dopredu inak prihrá loptu na náhodnú pozíciu dáždnika pomocou správania prihrávka na náhodnú pozíciu dáždnika.

Závislosti:

objekty: `world` (získovanie informácií o pozícií, o číslach atď.)

funkcie: `PlayerSkills::PassToStrategicPosition()`
`PlayerSkills::DumbKick()`
`PlayerSkills::PassToRandomUmbrellaPosition()`

Prihrávka na náhodnú pozíciu dáždnika

Algoritmus:

Zabezpečuje metóda `PlayerTactics::PassToRandomUmbrellaPosition()`, náhodne sa vyberie jeden bod z poľa `umbrellaPositions` a do tohto bodu hráč vystrelí pomocou metódy `PlayerSkills::DumbKick` (identifikované ako správanie kopnutie).

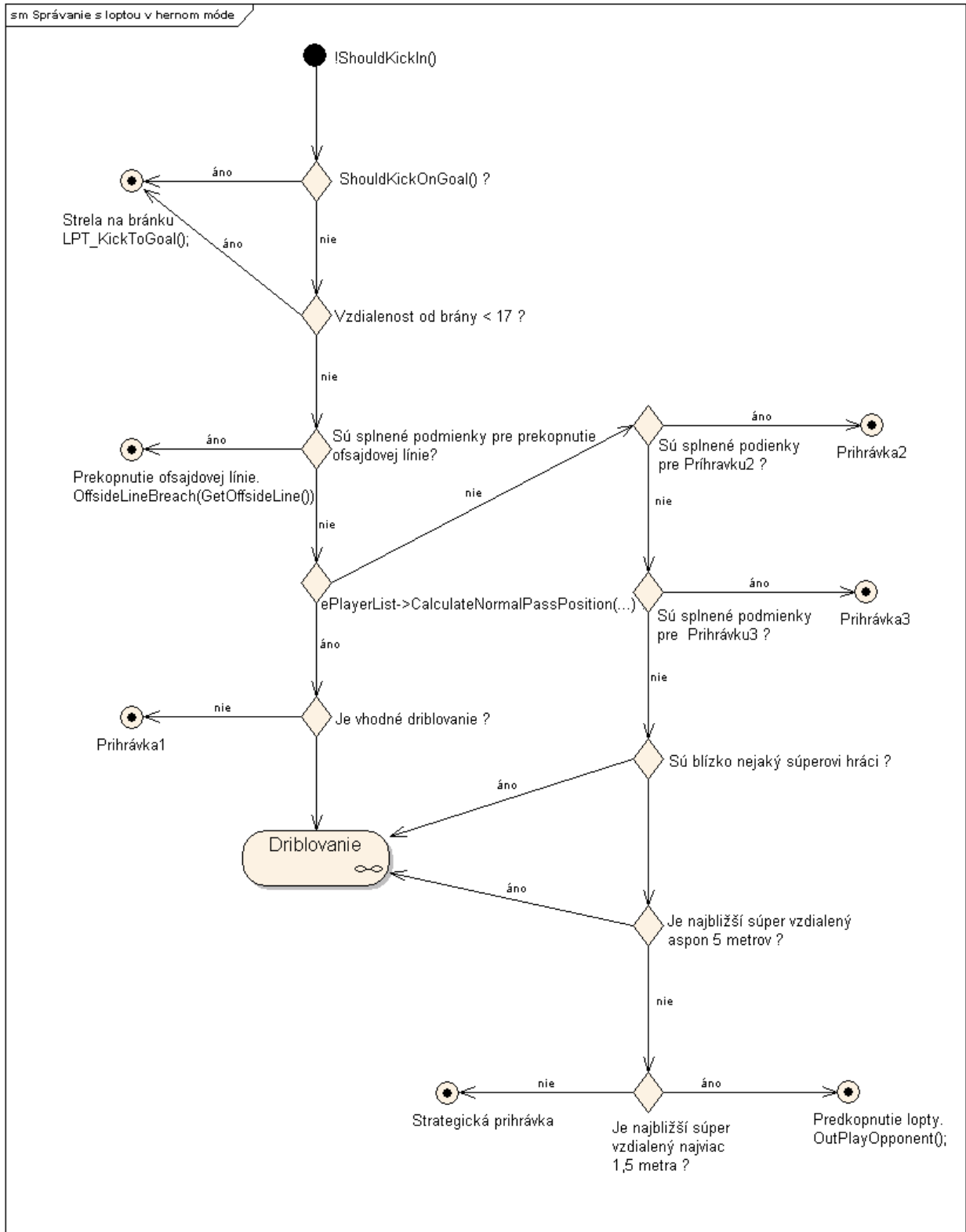
Závislosti:

objekty: `world` (získovanie informácií o pozícií, o číslach atď.)

funkcie: `PlayerSkills::DumbKick()`

Na obrázku (Obr. 9) je zobrazený kompletný stavový diagram pre hráča s loptou v hernom móde.

Analýza existujúcich hráčov



Obr. 9 - Stavový diagram správania hráča čakajúceho na rozohrávku

2.4.3 Analýza funkcií komunikujúcich so serverom

Základnou triedou, ktorej úlohou je komunikácia so serverom, je trieda `ServerSocket` definovaná v súboroch `./Common/Protocol/ServerSocket.{h|cpp}`. Nasledujúce metódy slúžia na pripojenie na server a prácu so správami:

- `SetServer(const char* addr, unsigned port)` je zodpovedná za pripojenie na server, parametrami sú adresa a port servera.
- `SendMsg(const char* msg)` odosiela správu na server.
- `ReceiveMsg()` prijíma správu zo servera a uloží ju do buffra. Potom sa správa parsuje.

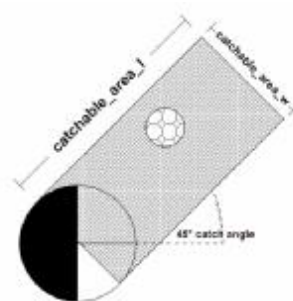
Podľa toho, akého je správa typu, sa spustí príslušný parser. Parser je definovaný v súboroch `./Common/Protocol/Parser.{h|cpp}`. Ak sa správa správne spracuje, zavolá sa príslušná metóda rozhrania. Pracuje s tromi rozhraniami: pre synchronizáciu, pre zmenu formácií alebo pre nastavenie premenných.

Odosielanie požiadaviek

Trieda `CommandGen` (`./Common/Protocol/CommandGen.{h|cpp}`). využíva triedu `ServerSocket` a generuje príkazy pre server. Obsahuje metódy umožňujúce ovládanie tela hráča a zisťovanie informácií o jeho tele. Správy sú odosielané na server vo formáte, aký je popísaný v prílohe Protokol pre komunikáciu so serverom. Názvy sú zväčša podobné, teda napr. metóda `Say()` posiela správu (`say ...`) s príslušnými parametrami.

Metódy v tejto triede:

- `ChangeView(ViewWidth iWidth)` slúži na zmenu uhlu pohľadu. Táto funkcia prijíma iba jeden parameter, šírku pohľadu (`NARROW_VIEW`, `NORMAL_VIEW` alebo `WIDE_VIEW`) a implicitne posiela správu na server s parametrom kvality pohľadu `high`. Frekvencia získavania informácií o pohľade sa vypočítava zo šírky a kvality pohľadu. Zmena kvality pohľadu na `low` zvýši frekvenciu získavania informácií o pohľade, bolo by zaujímavé zistiť prečo je implicitne použitý parameter `high` a prípadne upraviť túto metódu.
- `Score()` odošle na server požiadavku na zistenie aktuálneho stavu zápasu.
- `Catch(float fDirection)` umožňuje brankárovi zachytiť loptu, ktorá sa nachádza v ohraničenom území pod uhlom ϕ ako je znázornené na obrázku (Obr. 10). Príkaz odosiela uhol, pod ktorým sa nachádza územie, kde sa má zachytiť lopta.



Obr. 10 - Územie kde brankár môže zachytiť loptu

- `Dash(float fPower)` zrýchli pohyb hráča v smere jeho tela. Parametrom je veľkosť zrýchlenia, ktorá môže nadobúdať hodnoty z intervalu `<minpower, maxpower>`.

- Ak hráč zrýchľuje smerom dopredu, jeho stamina sa zníži o veľkosť zrýchlenia. Ak hráč zrýchľuje dozadu, stamina sa zníži o dvojnásobok zrýchlenia. Ak hráč už nemá dostatok staminy na zrýchlenie, použije sa len toľko, koľko je dostupné.
- *Kick(float fPower, float fDirection)* kopne loptu s danou silou v danom smere. Sila kopnutia môže nadobúdať hodnoty <minpower, maxpower>. Príkaz *kick* bude vykonaný na serveri, ak má lopta stav kick-able pre tohto hráča a hráč sa nenachádza v ofsajde.
 - *Move(float fX, float fY)* presunie hráča priamo na danú pozíciu v poli. Je dostupná len na začiatku polčasu alebo pri góle. Je dostupná aj pre brankára, ak chytí penaltu. Môže sa pohybovať v šestnástke.
 - *Tackle(float power)* je šmýkačka, parametrom je sila šmýkačky. Hráč po šmýkačke musí 10 cyklov čakať kým môže pokračovať v hre.
 - *Turn(float fMoment)* otočí hráča o daný moment. Ak sa hráč nepohybuje, uhol, o ktorý sa otočí je rovný momentu. Ak je hráč v pohybe, uhol otočenia sa určí pomocou vzorca. Pri väčšej rýchlosti je uhol otočenia menší.
 - *TurnNeck(int iAngle)* otočí hlavou hráča o špecifikovaný uhol. Tento uhol je relatívny k telu hráča. Uhol hlavy hráča je uhlom, ktorým sa hráč pozerá. Môže byť vykonaný v rovnakom cykle ako otočenie, zrýchlenie, kopnutie.
 - *Say(const char* pMsg)* odošle správu ostatným hráčom, ktorí sa nachádzajú na území do určitej vzdialenosti (štandardne 50). Túto správu dostanú od servera všetci hráči v tomto území, teda aj súperovi (v dokumentácii Loptošov je funkcia pre odoslanie správy chybné označená ako *Send*).
 - *AttentionTo(int uniformNumber)* spôsobí, že ak hráč môže dostať správu od viacerých hráčov, nedostane náhodnú správu, ale správu od hráča, ktorého nastavil v tejto metóde. Správa odoslaná na server má formát: (*attentionto <TEAM> <UNUM>*). Metóda *AttentionTo* umožňuje nastaviť odposluch iba vlastného hráča, parametrom je jeho číslo.
 - *SenseBody()* pošle požiadavku na server o získanie momentálneho fyzického stavu hráča. Formát správy odoslanej serverom na túto požiadavku možno nájsť v prílohe Protokol pre komunikáciu so serverom. Informácie o stave tela sa automaticky posielajú v pravidelných intervaloch hráčom.

Prijímanie a spracovanie správ zo servera

Ako bolo spomenuté vyššie, po spracovaní správy sa predá jednému z rozhraní. Taktické rozhranie (*./Player/Interfaces/Tactics.h*) rozhoduje o vykonaných akciách.

- *ChangeFormation(const char* formation)* sa volá, ak príde správa na zmenu formácie. Metóda je implementovaná v *./Player/PlayerTactics.cpp*.

Pamäťové rozhranie (*./Player/Interfaces/Memory.h*) slúži na ukladanie informácií o prostredí hráča. Ak príde správa na nastavenie parametra, správa sa parsuje metódou *ParseVariable(const char *name, const char *value)* implementovanou v *./Player/World/Variables.cpp*. V súčasnosti toto rozhranie funguje len na zapnutie/vypnutie neurónovej siete.

Rozhranie pre synchronizáciu (*./Player/Interfaces/Synchronization.h*) rozhoduje o tom kedy sa vykoná akcia. Toto rozhranie spracováva aj informácie zo senzorov hráča (zrak, sluch a telo).

- *OnHear(int time, float direction, bool ourTeam, int uniform, const char* msg)* spracováva zvukové vnemy, ktoré hráč dostáva zo serveru ako správu (*hear time direction TEAM [UNUM] "message"*). Metóda je implementovaná

Analýza existujúcich hráčov

v ./Player/PlayerSkills.cpp. Parametre, ktoré metóda dostáva sú čas, smer odkiaľ prišla správa, či sa jedná o správu nášho tímu, číslo hráča a samotná správa.

- OnSee(const VISUAL_INFO*) spracováva vizuálne informácie zo správy prijatej od servera (see ...) . Metóda je implementovaná v ./Player/PlayerKernel.cpp. Vizuálne informácie prichádzajú hráčovi pravidelne v určitom intervale (štandardne 150 ms).
- OnSenseBody(const SENSE_BODY*) spracováva informácie o súčasnom fyzickom stave hráča. Server posielala informácie hráčovi ako správu (sense_body ...). Jej formát je možné nájsť v prílohe Protokol pre komunikáciu so serverom. Tieto správy sú automaticky posielané hráčovi v určitom intervale (štandardne 100 ms).

Ďalším typom spracovávaných informácií sú správy zo servera, ktoré posielala trénerovi. Ak sa klient pripojí na server, obdrží nasledujúce správy:

- (server_param ...)
- (player_param ...)
- (player_type ...)

Správu o parametroch servera a parametroch hráča dostane klient len raz, správu o type hráča dostane raz pre každý typ hráča. Nasledujúce metódy majú za úlohu spracovanie týchto správ:

- OnServerParam(const SERVER_PARAMS* params) vlastne len nastaví informácie o parametroch servera v modele sveta. Implementovaná v ./Player/PlayerKernel.cpp
- OnPlayerParam(const PLAYER_PARAMS* params) nastaví informácie o parametroch hráča v modele sveta. Implementovaná v ./Player/PlayerKernel.cpp
- OnPlayerType(int id, const PLAYER_TYPE* type) nastaví informácie o type hráča v modele sveta. Implementovaná v ./Player/PlayerKernel.cpp

Ak sa klient rozhodne dostávať vizuálne informácie v každom cykle, (*eye on*), bude dostávať nasledujúcu správu každých 100 ms: (*see_global (OBJ1 OBJDESC1) (OBJ2 OBJDESC2) ...*). Túto správu dostáva iba tréner, metódu pre spracovanie tejto správy má implementovanú *on*.

Informáciu o stave zápasu ako odpoveď na príkaz (*score*) spracúva metóda:

- OnScore(int time, int our, int their) implementovaná v ./Player/PlayerKernel.cpp. Nastaví informáciu o stave zápasu v modele sveta.

Zmena hracieho módu je oznámená rozhodcom. Rozhodca oznamuje aj ďalšie udalosti, ako *faul*, *gól*, *atď.* Tieto udalosti prichádzajú zo servera ako správa (*referee String*), kde *String* je určitý hrací mód alebo správa od rozhodcu. Zoznam hracích módov a správ od rozhodcu možno nájsť v prílohe Protokol pre komunikáciu so serverom. Tieto správy spracúva nasledujúca metóda:

- OnChangeMode(int time, PlayMode mode) je implementovaná v ./Player/PlayerKernel.cpp a je preťažená v ./Player/PlayerTactics.cpp aj v ./Player/PlayerSkills.cpp. Metóda v triede PlayerTactics sa používa pri trénovaní.

2.5 Zhodnotenie analýzy

Naším cieľom na predmete Tímový projekt je vylepšiť jeden z dostupných tímov. Z tohto cieľa vyplýva naša analýza, v ktorej sme sa zamerali na analýzu dostupných fakultných tímov a na hlbšiu analýzu jedného z nich. Aj kvôli odporúčeniu nášho vedúceho sme sa rozhodli modularizovať tím Loptoši, ktorý vznikol o rok pred nami. Analyzovali sme aj jeho predchodcu FIITBA a stručne aj ďalších predchodcov zo stromu tímov: FC Farmári, Stjupid Dox a Roztoče. Naším ďalším cieľom bolo kód tímu Loptoši modularizovať podobným spôsobom ako má 3D tím 6th Sense, ktorý je založený na správaniach a preto sme sa zamerali aj na analýzu tohto tímu.

Následne sme sa snažili hlbšie analyzovať kód a správania tímu Loptoši. V dokumentácii aj v kóde sme identifikovali rôzne správania, stručne ich charakterizovali a našli sme vzťahy medzi nimi. Správania sme si rozdelili na dve základne skupiny podľa toho, či hráč má alebo nemá loptu. Táto analýza správania bude slúžiť ako podklad pri modularizácii. Analyzovali sme aj kvalitu kódu, pri ktorej sme odhalili niektoré nedostatky, ktoré by bolo vhodné odstrániť. Zamerali sme sa aj komunikáciu hráča so serverom, pri ktorej sme identifikovali rôzne základné správania, ktoré budú využívať správania na vyššej úrovni.

3 ŠPECIFIKÁCIA

Problematikou robotického futbalu sa FIIT zaoberá približne osem rokov, a teda osem rokov sa vyvíjajú hráči simulovaného futbalu. Tento projekt je preto značne komplexný a modularizovať správanie hráča je preto náročné. Pre opísanú obtiažnosť jednotlivých stanovených cieľov nemáme za cieľ navrhovať a implementovať nové správania.

V našom projekte sa budeme primárne venovať normálnemu hráčovi. Ako vidieť z analýzy kódu, celé správanie je implementované v troch základných triedach (PlayerKernel, PlayerTactics, PlayerSkills). Problém týchto tried spočíva v tom, že tieto triedy sú veľmi veľké a je ťažko v nich identifikovať správania. Preto je ťažké, prakticky nemožné dopĺňať nové správania, respektíve obmieňať staré. Z toho dôvodu nie je možné ďalej rozvíjať a zlepšovať tohto hráča.

Našou snahou je identifikovať jednotlivé správania hráča v zdrojových súboroch (tak isto aj v dokumentácií) a popísať ich. Ďalej je potrebné identifikované správania naimplementovať tak, aby boli relatívne ľahko modularizovateľné. Toto si vyžaduje rozbiť základne triedy na menšie triedy tak, aby tieto triedy predstavovali jednotlivé identifikované správania. Tým dosiahneme jednoduchšie a prehľadnejšie triedy (správania). Tieto novovytvorené správania by sa dali ďalej medzi sebou kombinovať a tým vytvárať nové správania (niečo podobné ako na návrhu tímu 6th Sense na (Obr. 1).

Ďalej chceme v zdrojových kódach nájsť všetky funkcie, ktoré priamo komunikujú so serverom, a tak isto identifikovať dáta, ktoré sú posielané na server a prijímané zo servera. Z týchto funkcií je potrebné vytvoriť samostatný plne funkčný modul, ktorý bude slúžiť ako základ pre tvorbu nových správání.

4 HRUBÝ NÁVRH

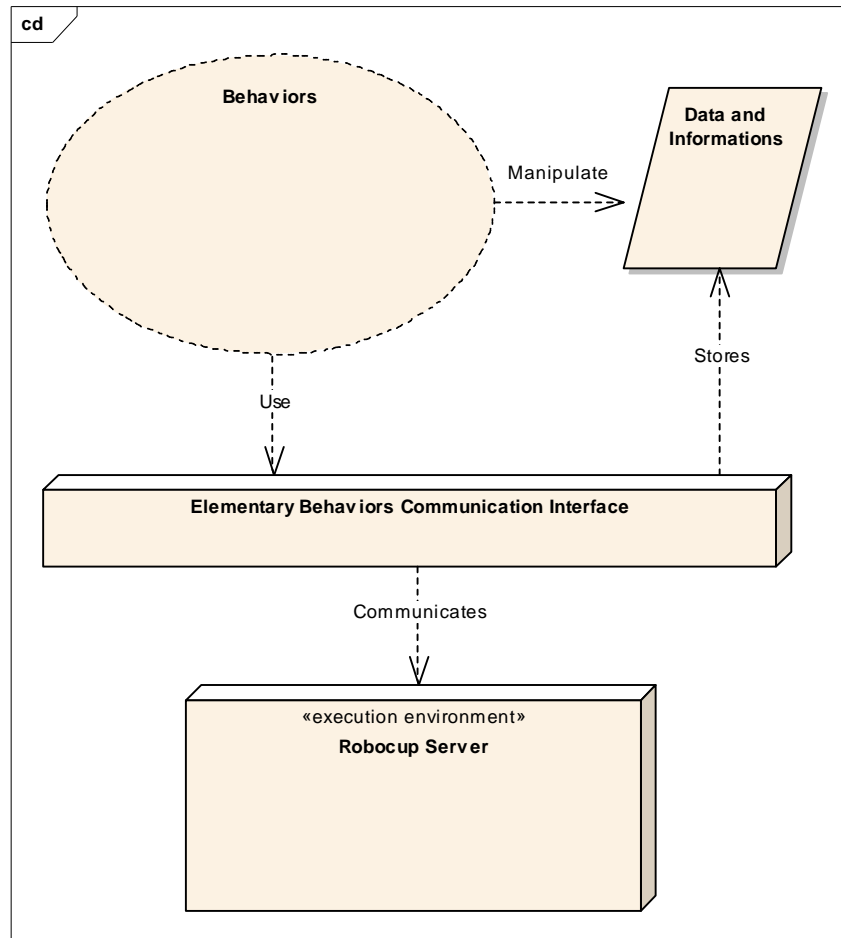
4.1 Refaktorizácia a vytiahnutie správanií

Našou snahou aj a cieľom je refaktorizovať zdrojový kód a vytiahnuť správania tak, aby bol do budúcnosti ľahko modularizovateľný. Jedno správanie môže pozostávať z viacerých správanií. A tieto menšie správania môžu byť ďalej pozostávať s ešte menších správanií. Takýmto postupným delením sa dostaneme až k najmenším elementárnym správaniam, ktoré sa už nedajú ďalej deliť. Tieto elementárne správania (*presunSaNaMiesto()*, *kopNaMiesto()*, *otocKrk()*...) realizujú samotné volania servera.

Každé správanie, či už vyššie alebo nižšie v konečnom dôsledku na najnižšej úrovni komunikuje so severom, ktorému posíla správy a prijíma správy zo servera. Predmetom týchto správ sú rôzne údaje o stave sveta, hráčoch, skóre a pod.

Základ našej architektúry vidieť na obrázku (Obr. 11). Za celý priebeh simulovaného zápasu je zodpovedný Server, ktorý prijíma a spracováva správy. Komunikácia so serverom prebieha cez modul, ktorý sprostredkováva správaniam elementárne funkcie. Správanie všeobecne pre svoju činnosť potrebuje vedieť dve veci. Prvou sú aktuálne informácie na základe, ktorých sa rozhoduje akú akciu vykonať. Tieto informácie predstavujú stav sveta, hráča, ale aj ďalšie informácie spoločné a potrebné pre fungovanie správanií. Také to informácie spoločné pre všetky moduly sa budeme tiež snažiť vytiahnuť von z tried a držať ich pohromade (Obr. 11).

Hrubý návrh



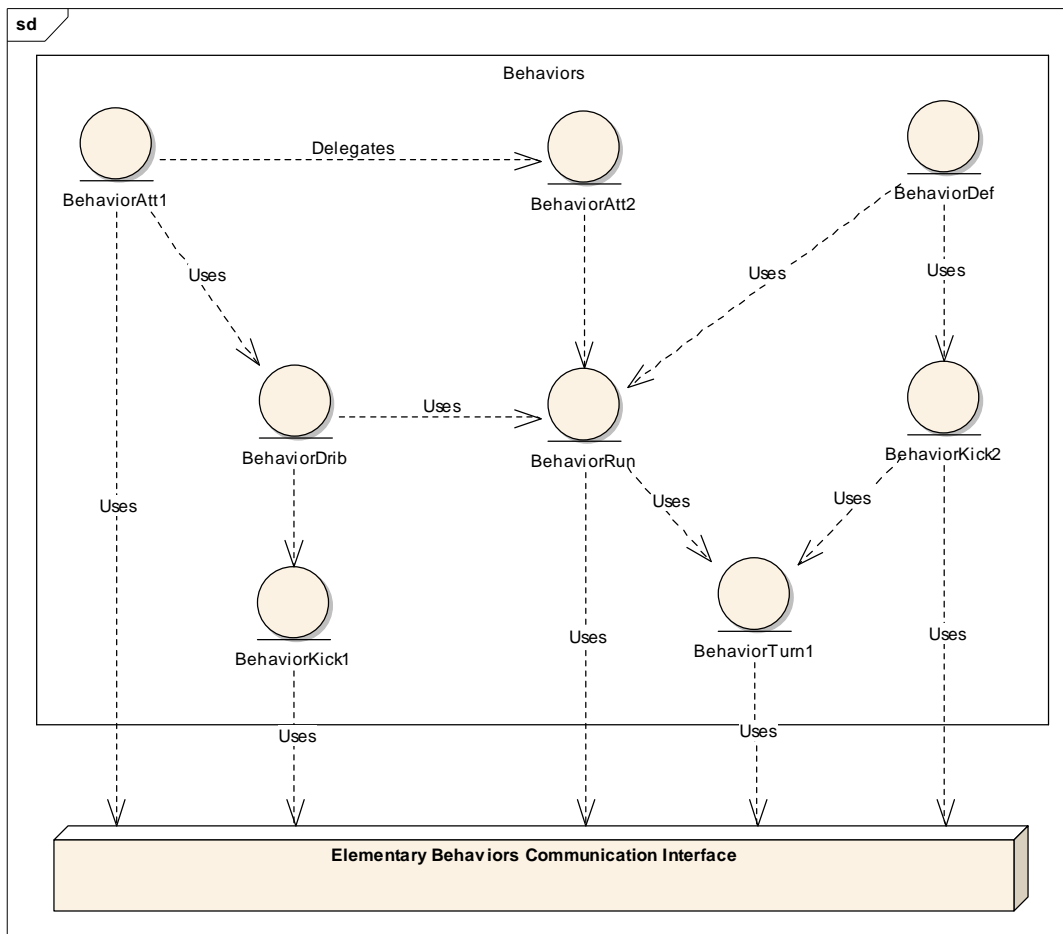
Obr. 11 - Základný návrh architektúry.

Akcie, ktoré sú výsledkom rozhodnutia správania môžu byť rôzne:

- správanie môže priamo zavolať modul, ktorý posiela správu na server
- správanie môže ponechať rozhodovanie na iné správanie, z ktorého je vytvorené
- správanie odovzdať rozhodovanie inému správaniu.
- správanie môže vykonať nič

Podľa takejto viazanosti správání (Obr. 12) nie je jednoduché rozdeliť správania do vrstiev a o to sa ani nesnažíme.

Hrubý návrh

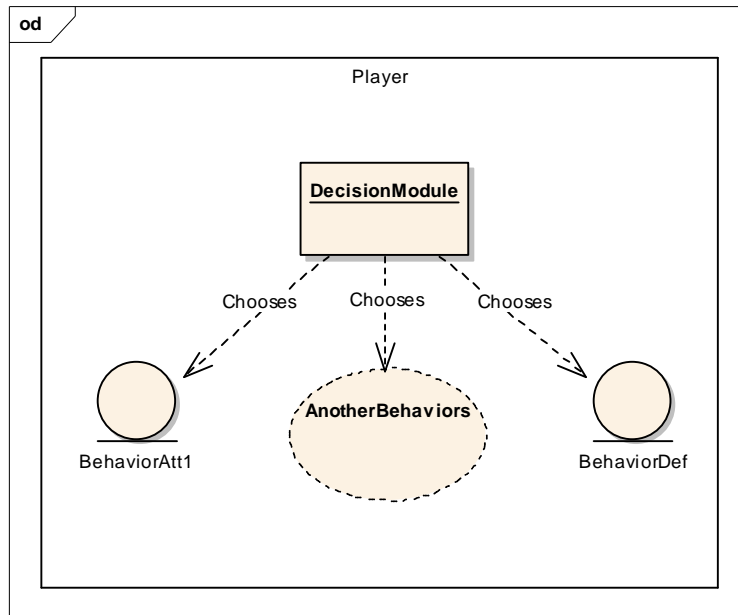


Obr. 12 - Spolupráca správání

Zoberme si hráča Loptošov, tak ako je implementovaný. Tento hráča má implementované hlavné metódy „správanie s loptou a správanie bez lopty.“ Z každej tejto metódy sú ďalšie funkcie ako „kopni do lopty, bež na miesto, otoč sa....“

Celý systém týchto metód tvorí jedno veľké správanie jedného hráča. Môžeme si to predstaviť ako „najvyššie“ správanie. Čiže tento jeden hráč má jedno najvyššie správanie, ktoré je zložené z menších správání. Celé rozhodovanie (kedy a ktorá metóda sa vyberie) je napevno nastavené a ťažko meniteľné. Povedzme, že nám toto „najvyššie“ správanie nevyhovuje, lebo je príliš defenzívne a chceme ho prerobiť na útočnejšie, ale zároveň chceme ho mať aj k dispozícii. To by sa dá vyriešiť tak, že sa zavedie modul, ktorého úlohou bude rozhodovať, ktoré z týchto „najvyšších“ správání vyberie (Obr. 13).

Tak isto by sa dal spraviť rozhodovací modul nad hocjakými rovnakými správáním napríklad pri výbere strely: „kopSilný, kopSlabý, kopStredný...“ V podstate by sa dalo povedať, že aj samotný rozhodovací modul je správanie.

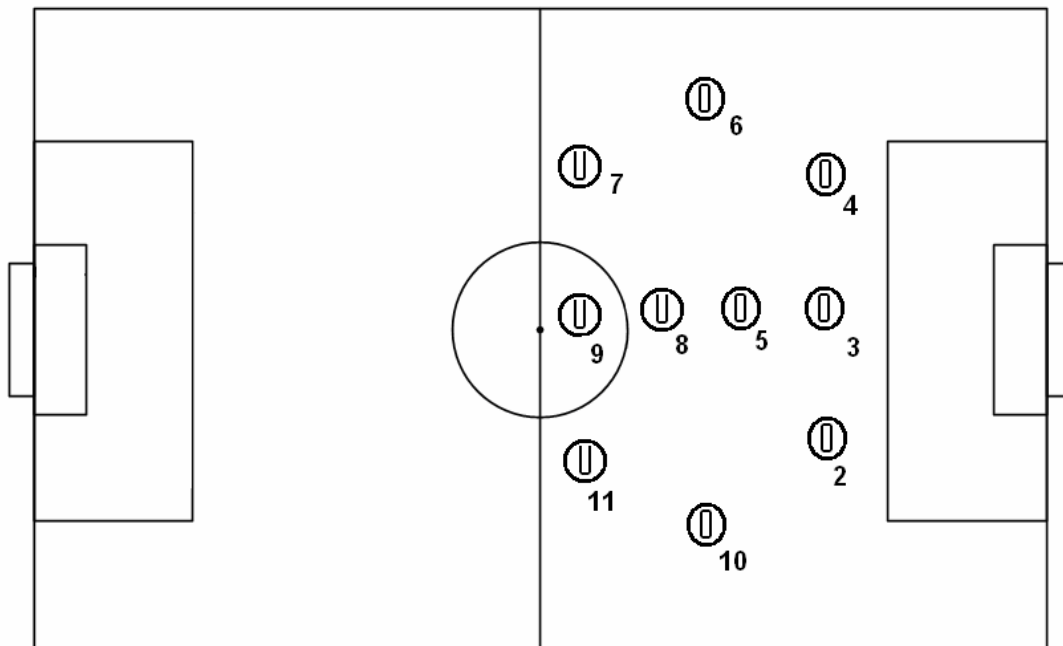


Obr. 13 - Rozhodovací modul hráča

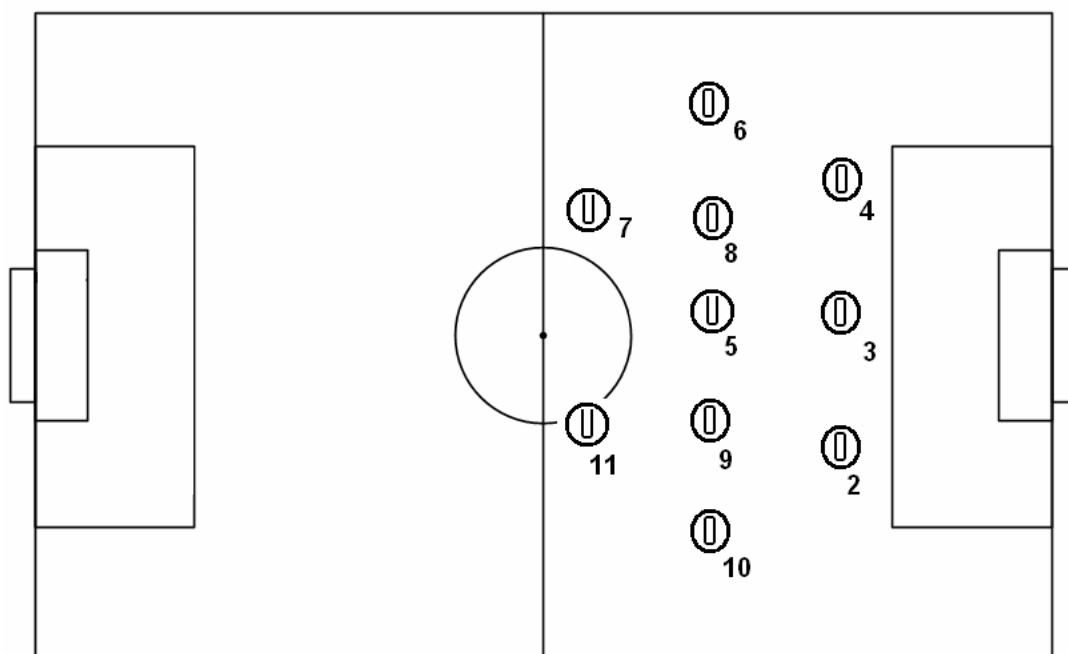
4.2 Loptošovská „Konceptia vedúceho hry“ a zmena správania

Mali by sme implementované dve „najvyššie“ správanie jedno útočné a jedno defenzívne. Čiže jeden hráč sa vie správa podľa dvoch správania útočnejšie alebo obrannejšie a počas hry je schopný si toto správanie meniť. Týmto sa dá nadviazať na prácu Loptošov. Loptoši zaviedli do hráča „Konceptiu vedúceho hry,“ ktorá je založená na zmene formácií a posielania správ. Pri ich zmene formácií dochádza len k zmene formácií, pričom správanie hráčov sa nemení. Doplnením rozhodovacieho modulu, ktoré by bolo schopné meniť najvyššie správanie by sa dosiahol lepšie usporiadaný a silnejší tím. Hráči by si vedeli zmeniť svoje „najvyššie“ správanie podľa toho kde vo formácií sa nachádzajú (Obr. 14 a 15).

Hrubý návrh



Obr. 14 - Zmena formácie so zmenou správania - pre zmenu formácie (U = útočné správanie, O = obranné správanie, 2..11 = čísla hráčov)



Obr. 15 - Zmena formácie so zmenou správania – po zmene formácie (U = útočné správanie, O = obranné správanie, 2..11 = čísla hráčov)



Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Ilkovičova 3, 842 16 Bratislava 4



Tvorba softvérového systému v tíme
Robocup – nové stratégie
Dokumentácia k riadeniu projektu

Tím č.6 – UTTP:

Bc.Peter Kajsa, Bc. Michal Kasan, Bc.Branislav Kontúr,
Bc. Dávid Kováč, Bc. Martin Nepšinský , Bc.Tibor Sekereš

Študijný program: Softvérové inžinierstvo/Informačné systémy

Ročník: 1, Inžinierske štúdium

Email: team06@googlegroups.com

Web: <http://www2.dcs.elf.stuba.sk/~team06is-si/>

Vedúci projektu: Ing. Marián Lekavý

Ak. rok: 2007/2008

| | | |
|----------|---|------------|
| 1 | ÚVOD | 1-1 |
| 2 | PONUKA | 2-1 |
| 2.1 | ČLENOVIA TÍMU | 2-1 |
| 2.2 | MOTIVÁCIA..... | 2-4 |
| 2.3 | ŠPECIFIKÁCIA A HRUBÝ NÁVRH RIEŠENIA | 2-4 |
| 2.4 | ĎALŠIE MOŽNOSTI A ROZŠÍRENIA | 2-5 |
| 2.5 | ZDROJE..... | 2-6 |
| 2.6 | PREFERENCIE TÉM TÍMU | 2-7 |
| 3 | PLÁN PROJEKTU..... | 3-1 |
| 4 | ROZDELENIE ÚLOH V TÍME..... | 3-1 |
| 4.1 | DLHODOBÉ ROLY | 4-1 |
| 4.2 | KRÁTKODOBÉ ROLY | 4-2 |
| 4.3 | ROZDELENIE ÚLOH V TÍME..... | 4-2 |
| 5 | KOMUNIKÁCIA V TÍME..... | 5-1 |
| 5.1 | PODPORA RIADENIA PROJEKTU..... | 5-1 |
| 5.2 | KOMUNIKÁCIA | 5-1 |
| 5.3 | ZDIELANIE ZDROJOV..... | 5-2 |

PRÍLOHY

PRÍLOHA A– ZÁPISY ZO STRETNUTÍ TÍMU

PRÍLOHA B– PREBERACIE PROTOKOLY

1 ÚVOD

V dnešnej dobe tvorba softvérového alebo informačného systému si vyžaduje prácu väčšieho počtu ľudí a nie jednotlivca. Je to dané najmä rozsiahlymi projektmi a použitím viacerých technológií. Z toho dôvodu je nutné, aby sa všetci nevenovali všetkému a ničomu, ale aby sa špecializovali na danú oblasť. Tu sa vynára otázka riadenie tímu, ktorá patrí k jednej z najdôležitejších častí úspešného vyriešenia projektu. Každý väčší projekt si vyžaduje takéto riadenie, ktorého úlohou je koordinovať a sledovať činnosť ľudí v tíme a tak isto aj komunikáciu ľudí v tíme.

V tomto dokumente sa nachádzajú dokumenty súvisiace s riadením projektu tímu číslo 6, ktorému bola pridelená téma „Robocup - nové stratégie“. V kapitole 2 sa nachádza ponuka vypracovaná tímom číslo 6. V kapitole 3 je možné nájsť plán projektu. Pridelenie úloh jednotlivým členom sa nachádza v kapitole 4. Kapitola 5 je venovaná komunikácii v našom tíme. V kapitole 6 sú všetky zápisy zo stretnutí v chronologickom poradí.

2 PONUKA

Táto časť dokumentu obsahuje ponuku tímu č. 6 UTTP vypracovanú k téme „HASIČI“. Obsahuje všetky časti, ktoré majú byť súčasťou ponuky.

2.1 Členovia tímu

Bc. Peter Kajsa (pyotr@ynet.sk)

Absolvoval som bakalárske štúdium v odbore Informatika študijný program Informatika na FIIT a momentálne som študentom študijného programu Softwarové inžinierstvo. Svoju bakalársku prácu som vypracoval na tému „Podpora návrhových vzorov v CASE nástrojoch“, kedy som sa oboznámil s mnohými návrhovými vzormi so skupiny GoF a zároveň som získal zaujímavé skúsenosti s prácou s CASE nástrojmi a s návrhovými vzormi. Mám praktické skúsenosti s technológiou Java EE a niektorými príslušnými aplikačnými rámcami implementujúcimi architektonické a návrhové vzory ako napríklad Struts. Veľmi dobré skúsenosti mám taktiež s jazykom PHP a príslušným aplikačným rámcom implementujúcim vzor MVC, CAKE. Pomocou uvedených technológií, predovšetkým však pomocou PHP, CAKE som vytvoril v minulosti mnohé reálne nasadené webové aplikácie rozmanitého druhu. Pomerne rozsiahle skúsenosti mám taktiež s tvorbou grafických návrhov(CSS, XHTML, JavaScript) a s navrhovaním a prácou s databázami.

Bc. Michal Kasan (michal.kasan@gmail.com)

Venujem sa predovšetkým tvorbe webových aplikácií so zameraním na technológiu JSP, platformu Java EE a využitie frameworkov Struts a Hibernate. Ako java-programátor som pracoval na dvoch interných projektoch medzinárodnej spoločnosti. Oba projekty boli webové aplikácie vytvorené na platforme Java EE. Obe aplikácie boli nasadené na aplikačnom serveri Apache Tomcat a ich údaje boli spravované databázovým systémom Microsoft SQL Server. Na správu verzií zdrojových súborov aplikácií som používal CVS.

V rámci svojej bakalárskej práce som špecifikoval, analyzoval, navrhoval, implementoval a testoval rozsiahlejší softvérový systém, webovú aplikáciu, ktorá demonštrovala využitie architektonických vzorov v praxi. Pri vývoji softvérového systému som postupoval podľa metodiky Unified process. Vďaka tomu som si osvojil detaily pracovných postupov v jednotlivých etapách vývoja softvérového systému. Všetky vyššie spomenuté technológie boli použité pri implementácii tohto systému. V rámci tímu plánujem zastávať rolu analytika a návrhára informačného systému, keďže som sa v tejto pozícii viackrát ocitol. Moje zameranie taktiež dokladuje výber jedného z povinne voliteľných predmetov a to „Architektonické a návrhové vzory pre programové informačné systémy“. Ako študent bakalárskeho štúdia na Fakulte informatiky a informačných technológií som absolvoval predmety Databázové systémy, Princípy softvérového inžinierstva. Získané znalosti z týchto predmetov mi môžu

výrazne napomôcť pri tvorbe informačného systému tohto typu. V súčasnosti nie som zamestnaný, preto sa môžem plnohodnotne venovať tímovému projektu.

Bc. Branislav Kontúr (bkontur@gmail.com)

Počas štúdia na bakalárskom študijnom odbore Informatika som získal základné skúsenosti s programovacím jazykom Java a SQL. Téma bakalárskeho projektu bola: "EHealth: Wireless Wellness Monitoring". Téma bola zameraná na využitie bezdrôtových technológií a GPS pri bezdrôtovom sledovaní zdravotného stavu pacientov. Práca bola zameraná na spracovanie GPS a nameraných údajov a posielanie rôznych druhov notifikácií (SMS, e-mail, interné správy a iné). Výstup práce bola séria webových aplikácií komunikujúca s databázovým serverom MySQL a SMTP serverom pre posielanie e-mailov.

Od 13. novembra 2006 som zamestnaný ako Java Junior Programátor vo firme Tempest a.s v Divízií INDEV, ktorá sa zaoberá tvorbou portálových riešení postavených na platforme JEE. Podieľam sa na realizácii a podpore projektov pre najväčších slovenských operátorov (Orange, T-com).

Projekty, na ktorých participujem:

- **T-com** - www.station.sk - portál pre kupovanie hudby, hier, požičiavanie videa cez internet, WebCMS, posielanie SMS, billingový systém, reporting...
- **T-com** - zona.t-com.sk - **Credit** - platobný systém AnyPay (rôzne platobné metódy)
- **T-com** - zona.t-com.sk - **Credit** - Marketingový admin (nastavovanie cien, dĺžky, prevádzky... hovorov pre Credit a Hlas cez Internet)
- **T-com** - zona.t-com.sk - **Credit** - Webmail - Sun Java Communication Express - prispôsobovanie sunovského e-mailového klienta pre T-com
- **Orange** - PhotoBlog a Geopedia

Základné skúsenosti:

- Operačný systém Solaris
- Sun Java System Application Server 8.2 (9.0), Sun Java System Access Manager, Sun Portal Server 6.0, Tomcat
- Oracle databázy, MySQL, LDAP
- Java SDK, Java Enterprise Edition

Technológie, s ktorými som prišiel do kontaktu:

- servlety, portlety, JSP, HTML, CSS, XSLT, XML, JavaScript, DWR
- EJB 2.X, Spring, Spring MVC, Hibernate, JDBC, SQL
- Web-services (Axis)
- Maven, CVS

Bc. Dávid Kováč (qavidko@gmail.com)

Počas bakalárskeho štúdia na Fakulte informatiky a informačných technológií som si osvojil programovací jazyk Java (Java SE, Java EE). Popri štúdiu som sa zoznámil s technológiami PHP, HTML, CSS, SQL, Apache, CVS, Adobe Photoshop. Témou mojej bakalárskej práce bola „Anotácia on-line hypertextových dokumentov“. Pri vypracovávaní tejto bakalárskej práce som sa zdokonalil v technológiách JavaScript, HTML, CSS, XML, DTD, XUL, XPCOM. Popri štúdiu som pracoval v štátnej organizácii, kde som zastával pozíciu správcu systému. Momentálne však nie som zamestnaný, a tak sa môžem plnohodnotne venovať tímovému projektu. Počas štúdia som absolvoval predmety Databázové systémy, Princípy softvérového inžinierstva, Projektovanie aplikácií počítačov, z ktorých znalosti mi môžu napomôcť pri práci na tímovom projekte. Pri práci na tímovom projekte môžem prispieť testovaním systému a tiež ako programátor.

Bc. Martin Nepšínský (martin.nepsinsky@gmail.com)

Počas štúdia na bakalárskom študijnom odbore Informatika a počas práce vo firme Interway, s.r.o. som získal rôzne skúsenosti s vývojom informačných systémov na platforme Java EE, ktoré sa dajú využiť pri vývoji systému na oznamovanie požiarov dobrovoľným hasičom. V bakalárskom projekte: Vývoj pomocou modelom riadenej architektúry som vyvinul dva informačné systémy pre správu študentov a predmetov na univerzite s využitím najmodernejších a najrozšírenejších nástrojov na platforme Java EE. Pracoval som pri vývoji s rôznymi technológiami na všetkých vrstvách ako napríklad: Struts, JSF, JSP, CSS, JavaScript, Spring, EJB, Hibernate.

V práci som sa podieľal na vývoji redakčného systému WebJET, ktorý umožňuje automatické odosielanie zvolených dát vybraným používateľom pomocou SMS alebo e-mailu. Tieto znalosti by som rád využil pri oznamovaní požiaru vybraným dobrovoľným hasičom. Ďalej som sa zúčastnil na vývoji rôznych systémov na platforme Java EE využívajúc technológie: Struts, Stripes, Cayenne, JavaScript, JSP, servlety, Axis a iné, ale aj na mnohých menších aplikáciách na kľúč pre rôznych klientov.

Medzi najväčšie projekty, na ktorých som sa zúčastnil patria:

- Kompletne softvérové riešenie tipovania kurzových stávk na internete pre stávkovú kanceláriu VictoryTip
- On-line rezervačný systém zájazdov pre cestovnú kanceláriu Dekampo
- Redakčný systém WebJET
- Systém na monitorovanie správ v médiách pre Neopublic Porter Novelli
- Systém na manažment a evidenciu výkonnosti poradcov pre FinCo&Partners
- Systém na poskytovanie úverov pre ESSOX SK
- On-line poistenie a monitorovanie výkonnosti podielových fondov pre Generali Poist'ovňa, a.s.
- Systém na evidenciu poskytovateľov zdravotnej starostlivosti, zdravotných pomôcok a on-line prihlášku/odhlášku pre Všeobecnú zdravotnú poisťovňu, a. s.

Mám praktické skúsenosti s aplikáciami Apache Tomcat, jBoss, MySQL, PostgreSQL, Eclipse, MyEclipse a EMS SQL Manager.

Bc. Tibor Sekereš (sekino@zoznam.sk)

Absolvoval som strednú priemyselnú školu v Trnave, odbor elektrotechnika. Ďalej som študoval informatiku na Fakulte prírodných vied Univerzity sv. Cyrila a Metoda v Trnave. Bakalársku prácu som vypracoval na tému „GUI pre delenie šírky prenosového pásma v Linuxe“. Pri jej vypracovávaní som získal skúsenosti s programovaním v OS

Linux a s pomocou knižnice Qt. Mám skúsenosti s programovaním v prostredí C++Builder a vývojom multiplatformových aplikácií s knižnicou Qt, ďalej mám základné znalosti jazyka Java. Počas štúdia som sa oboznámil s jazykmi PHP, SQL, LISP. Z databázových systémov mám praktické skúsenosti s MySQL. Zaujímam sa o administráciu sietí založených na OS Linux/Unix.

Plán projektu, pre lepšiu vizuálnu predstavu o rozložení prác v čase, uvádzame okrem tabuľkovej formy aj vo forme Ganttovho diagramu. Jednotlivé termíny, z hľadiska výstupov projektu, sú v tabuľke plánu vyznačené tučným písmom.

2.2 Motivácia

Našou motiváciou pre prácu na tejto téme je získať skúsenosti s vývojom projektu väčšieho rozsahu a osvojiť si prácu v tíme. Hlavným dôvodom je však možnosť vytvárať systém, ktorý sa plánuje reálne nasadiť a využívať, vďaka čomu môžeme pomôcť pri ochrane života a zdravia ľudí, ako aj ich majetku. V budúcnosti bude užitočný pre veľkú skupinu ľudí a nakoniec aj pre nás samotných. Za ďalšie výhody tejto témy považujeme, že pre projekt nie je vopred špecifikovaná platforma systému a použité technológie, vďaka čomu môžeme využiť aplikačné rámce, ktoré budeme považovať za najvhodnejšie pre danú problematiku. Zaujímavé je aj využitie mobilných technológií a systému Google Maps, ktorý plánujeme použiť. Dôležitým faktorom pri výbere projektu bol taktiež fakt, že takmer všetci členovia tímu majú skúsenosti s webovými technológiami a tým aj veľký predpoklad vytvoriť projekt v požadovanej kvalite.

2.3 Špecifikácia a hrubý návrh riešenia

Špecifikácia funkcionality

- § rozosielenia notifikácií vybraným dobrovoľným hasičom
- § vytvorenie záznamu o požiari
- § evidencia hasičov (osobné údaje, miesto pobytu)
- § evidencia udalostí
- § sledovanie hasičov pomocou GPS a Google Maps

Ponuka

- § rozosielanie rôznych druhov notifikácií (SMS, hlasové správy, e-maily, správy na pager atď.)
- § vyhľadanie najbližších hasičov k miestu požiaru a zobrazenie požiarnikov na mape podľa GPS a podľa miesta požiaru.
- § GPS a real-time sledovanie hasičov

Automatické ohlasovanie

- § prepojenie s 112 (respektíve 150, 155, 158)
- § prerobenie na portálové riešenie (portálový server - portlety)

Vyhľadávanie najbližších hasičov:

- § statická – vyhľadávanie podľa miesta pobytu
- § dynamická – vyhľadávanie podľa GPS polohy (podmienka, osoba musí mať GPS zariadenie)
- § iné kritériá

Možný scenár:

- § operátor dostane hlásenie o požiare (Kde?)
- § vo vytvorenom systéme si dá vyhľadať uvedené miesto
- § systém mu zobrazí miesto požiaru a taktiež aj najbližších dobrovoľných hasičov (možnosť nastavenia rôznych heuristik pre vyhľadávanie)
- § operátor bude mať možnosť poslať notifikácie všetkým (vybraným) zobrazeným hasičom (zborom)

2.4 Ďalšie možnosti a rozšírenia

- § Vyhľadávanie hasičov nielen podľa GPS vzdialenosti (vzdušná čiara), ale pomocou nejakého systému, ktorý určí presnú trasu po ceste a dobu, za ktorú sa môže hasič dostať na miesto požiaru
- § E-mailové schránky pre hasičov v systéme
- § Komunikácia v rámci "portálu" (vymieňanie skúsenosti)
 - Rýchle správy
 - Chat
 - Blog
 - Diskusné fórum
 - Nástenka (príspevky, fotografie)
- § Organizácia a riadenie jednotlivých zborov – vedenie všetkých administratívnych záležitostí, hospodárenie, ekonomika.....
- § Ďalšia možná heuristika výberu hasičských zborov podľa príslušnosti k danému okresu plus susedných okresov - (bez použitia GPS)

Čo by sme chceli minimálne realizovať

- § Komunikácia GPS a serverov s hasičmi

- § Posielanie SMS na mobilné telefóny

Použiteľné softvérové technológie

- § **Front-end:** webové aplikácie – HTML, CSS, JSP, XSLT, XML, JavaScript, DWR
 - Struts (Spring MVC), Google API pre mapy
- § **Back-end** (servisná a databázová vrstva) – Spring, Hibernate, Web services (Axis), Java EE (EJB...), PL/SQL

Hardvérová infraštruktúra

- § Aplikačný server (webový, poprípade server s kontajnerom pre EJB)
- § Databázový server (Oracle, PostgreSQL, MySQL)
- § Message server

Mobilný telefón s podporou GPS

2.5 Zdroje

Minimálne jeden stroj = Development server + Databázový server

Development server (Softvér):

- § Java SDK 6.0
- Alternatíva 1
 - § Operačný systém: Solaris (Open-source)
 - § Aplikačný server: Sun Java System Application Server 9.0 - alternatíva Tomcat (nevýhoda nemá kontajner pre EJB)
- Alternatíva 2
 - § Operačný systém: Linux (Open-source)
 - § Aplikačný server: Tomcat
- § Diskový priestor: 4-8 GB
- § Pamäť: odporúčané minimum 512 MB

Databázový server:

- § Oracle
- § PostgreSQL
- § MySQL

Databázový klient:

- § EMS SQL Manager – verzia pre zvolený databázový server

Autentifikácia a autorizácia (voliteľné, nemusí byť, navyše)

- § OpenSSO (Single Sign On)
- § Sun Java System Access Manager
- § LDAP

Vývojové prostredie:

- § Eclipse 3.2 WTP – alternatíva NetBeans 5.5, (NetBeans 6 Beta)
- § CVS
- § IBM Rational software architect

Práca na projekte:

- § minimálne 13 hod za týždeň každý člen tímu
- § 150 hodín prvý semester – návrh, špecifikácia, implementácia
- § 250 hodín druhý semester – implementácia, nasadenie, testovanie

2.6 Preferencie tém tímu

1. Oznamovanie požiarov dobrovoľným hasičom (HASICI)
2. Báza znalostí a zručností študentov (ZNALOSTI)
3. Informačný systém pre komunikáciu s absolventmi (ALUMNI)
4. Podpora riadenia projektovo-orientovanej firmy (RIADENIE)
5. Tvorba rozvrhov (ROZVRH)

3 PLÁN PROJEKTU

Nižšie uvedená tabuľka číslo 1 zobrazuje plán činností, ktoré boli definované členmi tímu. Definovanie týchto činností vyplynulo z úloh zadaných pedagogickým vedúcim, pričom tieto činnosti vznikli rozbitím spomínaných úloh a ich časové ohraničenie odráža všetky termíny, ktoré je potrebné dodržať.

| týždeň | od-do | činnosť |
|--------|----------------------------|---|
| 3. | 11.10 – 17.10 | √ Oboznámenie sa zo zadaním. |
| | | √ Vytvorenie plánu. |
| | | √ Rozdelenie úloh v tíme. |
| | | √ Vytvorenie šablón pre dokumentáciu. |
| | | √ Začiatok analýzy domácich a zahraničných hráčov ROBOCUPU. |
| 4. | 18.10 – 24.10 | √ Vytvorenie webovej prezentácie. |
| | | √ Vyhládanie a zavedenie potrebných vývojových a podporných nástrojov. |
| | | √ Nástroj pre manažovanie a riadenie – TeamworkProject. |
| | | √ SVN - Subversion pre Eclipse. |
| | | √ IDE - Eclipse C++. |
| 5. | 25.10 – 31.10. | √ UML - Enterprise Architekt. |
| | | √ Spustenie prvého zápasu LOPTOSI vs. LOPTOSI. |
| | | √ Výber hráča pre refaktorizáciu. |
| | | √ Vytvorenie dokumentu analýzy domácich a zahraničných hráčov ROBOCUPU. |
| | | √ Zhrnutie analýzy |
| 6. | 1.11 – 7.11 | √ Identifikácia správania v zdrojových kódach a dokumentoch. |
| | | √ Vytvorenie dokumentu o identifikovaných správaniach. |
| | | √ Začiatok návrhu možných riešení. |
| 7. | 8.11 – 14.11 | √ Ukončenie tvorby návrhu riešenia. |
| | | √ Pripomienkovanie, kontrola a oprava dokumentácie. |
| | | √ Finalizácia prvej verzie dokumentácie. |
| 8. | 15.11 – 21.11 | √ Odovzdanie dokumentácie analýzy problému, špecifikácie požiadaviek a návrh riešenia |
| | | √ Vypracovávanie posudku analýzy, špecifikácie a návrhu pre konkurenčný tím. |
| | | √ Pripomienkovanie a kontrola posudku konkurenčného tímu. |
| | | √ Revízia na základe posudku. |
| | | √ Návrh prototypu a rozdelenie prác na prototypu. |
| 9. | 23.11. 21.11 – 27.11 | √ Odovzdanie posudku analýzy, špecifikácie a návrhu iného tímu |
| | | √ Ukončenie návrhu prototypu. |
| | | √ Začiatok implementácie prototypu. |
| 10. | 28.11 – 4.12 | √ Začiatok tvorby dokumentácie k prototypu. |
| | | √ Implementácia prototypu. |
| | | √ Testovanie a ladenie prototypu. |

plán projektu

| | | |
|-----|------------------|--|
| | | Tvorba dokumentácie k prototypu. |
| | | Testovanie a ladenie prototypu. |
| 11. | 5.12 – 11.12 | Ukončenie implementácie prototypu. Finalizácia dokumentácie k prototypu. |
| 12. | 12.12 – 18.12 | Pripomienkovanie a kontrola dokumentácie. Tvorba prezentácie k prototypu. |
| | 17.12 – 20.12 | Používateľská prezentácia prototypu |
| | 11.2. 2008 | Odovzdanie posudku prototypu iného tímu |

Tab. č. 1

4 ROZDELENIE ÚLOH V TÍME

Táto kapitola obsahuje pridelenie rolí, pričom člen tímu k danej role má priradené zodpovednosti a právomoci, ktoré mu v tejto funkcii prináležia. Nami stanovené role sú nielen dlhodobé, ale v priebehu projektu sa neskôr stanovia niektoré krátkodobé role, ktoré sa prispôsobujú potrebám projektu. Krátkodobé role môžu vzniknúť a zaniknúť a to podľa aktuálnej etapy projektu.

4.1 Dlhodobé roly

Vedúci tímu – Branislav Kontúr

Zodpovednosť:

- stanovenie poradia činností, ktoré treba vykonať
- koordinácia členov tímu
- kontrola termínov
- ako člen tímu musí produkovať výstupy

Manažér kvality – Martin Nepšínský

Zodpovednosť:

- kontrola kvality výstupov
- ako člen tímu musí produkovať výstupy

Manažér podporných činností – Dávid Kováč

Zodpovednosť:

- zabezpečuje fungovanie nástrojov na komunikáciu
- správca servera SVN
- rad ďalších podporných činností
- ako člen tímu musí produkovať výstupy

Manažér dokumentácie – Michal Kasan

Zodpovednosť:

- tvorba vhodných šablón pre dokumenty
- sledovanie termínov z pohľadu odovzdávaných dokumentov
- finalizovanie a kompletizácia dokumentov
- ako člen tímu musí produkovať výstupy

Web dizajnér – Peter Kajsa

Zodpovednosť:

- neustála aktualizácia stránky
- ako člen tímu musí produkovať výstupy

Vedúci vývojárov – Tibor Sekereš

Zodpovednosť:

- pridelovanie úloh týkajúcich sa zdrojových súborov
- neustále štúdium zdrojových súborov
- ako člen tímu musí produkovať výstupy

4.2 Krátkodobé roly

V ďalších fázach projektu sa náš tím určite nezaobíde bez

- hlavného architekta
- hlavného testera

4.3 Rozdelenie úloh v tíme

Na projekte náš tím doposiaľ pracoval na krátkodobých úlohách a na niektorých dlhodobých stále pracuje. Priradenie úloh k ich realizátorom je podrobne zachytené v prílohe A, zápisy zo stretnutí. Okrem stavu vykonania úlohy sú tu tiež termíny pridelenia úloh, termíny ukončenia a termíny skutočného ukončenia úloh.

Vychádzajúc z obsahu dokumentu „Inžinierske dielo“ bude v tejto časti opísané, kto jednotlivé tohto dokumentu realizoval. Tabuľka číslo 1 prehľadne zobrazuje relevantné informácie o rozdeľovaní úloh v našom tíme.

| Popis úlohy | Realizátor | Stav |
|--|------------|----------|
| Kapitola 1 Úvod | Kasan | vykonaná |
| Kapitola 2 Špecifikácia | Kontúr | vykonaná |
| Kapitola 3.1 Analýza 6th Sence | Kasan | vykonaná |
| Kapitola 3.2 Analýza FIITBA | Nepšinský | vykonaná |
| Kapitola 3.3 Analýza Brainstormers 2005 | Sekereš | vykonaná |
| Kapitola 3.4 Analýza Loptoši | Kajsa | vykonaná |
| Kapitola 4 Hrubý návrh | Všetci | vykonaná |
| Kapitola 5 Ďalšie požiadavky a ohraničenia | Všetci | zrušená |

Tab. č. 2

5 KOMUNIKÁCIA V TÍME

Jedna z najdôležitejších vlastností tímu je jeho schopnosť efektívne komunikovať problémy, ktoré vzniknú v rámci projektu. S tým je spojená rada nástrojov a technológií, ktoré túto efektívnu komunikáciu zabezpečia. V tejto kapitole budú teda opísané naše spôsoby komunikácie a nami používané nástroje zabezpečujúce komunikáciu. Okrem komunikácie členov tímu táto kapitola zachytáva riadenie projektu, ktoré je taktiež neoddeliteľnou súčasťou projektov.

5.1 Podpora riadenia projektu

Pre podporu riadenia projektu, pridelenie, sledovanie a plnenie úloh bol použitý „Teamwork project“, ktorý je dostupný na <http://online.twproject.com/>. Tento nástroj poskytuje pre náš tím všetku potrebnú funkcionálnu podporu. Neexistujú nijaké obmedzenia ani nevýhody tohto systému, ktoré by nám bránili v riadení projektu. Vedúci tímu definuje v nástroji úlohy, na nadchádzajúci týždeň pre jednotlivých členov tímu. V definícii úlohy je jej stručný opis spolu s termínom jej vykonania. Pridelenie „task-u“ je automaticky zasielané „e-mail-om“ danému členovi, ktorému bol „task“ pridelený. Na podobných princípoch pracuje viacero štandardných nástrojov určených na manažovanie projektu.

5.2 Komunikácia

Okrem nástroja Teamwork nám komunikáciu uľahčujú:

- E-mail
- Webové fórum
- Webová stránka
- Priebežné stretnutia s vedúcim tímového projektu
- Skype – Verbálna
- Instant Messaging (ICQ)
- Telefón

E-mail

Náš tím si vytvoril e-mailovú grupu <http://groups.google.com/group/TPteam06/>. V tejto grupe sú nielen členovia tímu, ale aj náš pedagogický vedúci. Tento druh e-mailu zasiela správy všetkým členom danej grupy. Pomocou jeho je vedená diskusia na rôzne témy, posielanie oznamov, zasielanie dôležitých informácií z hľadiska plnenia daných úloh.

Webové fórum

Tento druh komunikácie je výhradne určený na diskusie na rôzne témy. Taktiež je možné použiť fórum na vyjadrenie vlastného názoru a na poukázanie rôzne zaujímavosti spojené projektom. Stránka fóra je: <http://team06.wz.cz>.

Webová stránka

Tento druh komunikácie je výhradne určený na komunikáciu s okolitým svetom. Na stránke sú pravidelne aktualizované priebežné výstupy členov tímu. Používateľ tu nájde dôležité oznamy a hlavne dôležité odkazy(linky) súvisiace s projektom! Naša stránka je: <http://www2.dcs.elf.stuba.sk/TeamProject/2007/team06is-si/>.

Priebežné stretnutia

Jedná sa o pravidelné týždenné stretnutia s naším vedúcim Mariánom Lekavým. Sú to trojhodinové stretnutia, pričom účasť všetkých členov tímu je povinná. Z každého stretnutia sa vytvárajú pomerne podrobné zápisy, v ktorých sa nachádzajú všetky kľúčové body týkajúce sa stretnutí(úlohy na ďalší týždeň, závery). Zápisnica, kde sú všetky zápisy, je uvedená ako príloha A tohto dokumentu.

Iné spôsoby komunikácie členov tímu

Okrem všetkých vyššie uvedených spôsobov komunikácie náš tím siahla aj po štandardných spôsoboch komunikácie, ako sú:

- Skype
- Instant Messiging (ICQ)
- Telefón

5.3 Zdieľanie zdrojov

Na zdieľanie všetkých zdrojov kódov, dokumentácií a iných materiálov bol vytvorený repozitár, ktorý je dostupný na <https://uttp.googlecode.com/svn/trunk/UTTP>.

Repozitár je rozdelený na tri časti:

- docs – zdieľanie všetkých dokumentácií
- other – zdieľanie iných materiálov
- prog – zdieľanie zdrojových kódov

PRÍLOHA A – ZÁPISY ZO STRETNUTÍ TÍMU

Zápis zo stretnutia č. 1

Číslo stretnutia: 1

Dátum: 12.10.2007 13:00

Miesto: Softvérové štúdio (D07b)

Zapisovateľ: Michal Kasan

Overovateľ: Peter Kajsa

Prítomní:

Pedagogický vedúci: Ing. Marián Lekavý

Členovia tímu: Bc. Michal Kasan, Bc. Peter Kajsa, Bc. Dávid Kováč, Bc. Martin Nepšinský, Bc. Branislav Kontúr, Bc. Tibor Sekereš

Téma stretnutia

Vedúcim nášho tímu, Mariánom Lekavým, bolo navrhnuté, aby sme sa venovali refaktorizácii jedného z existujúcich hráčov na fakulte. Mali by ste teda zobrať niektorého z existujúcich hráčov a od základov prerobiť jeho štruktúru tak, aby boli jednotlivé časti hráčov (na úrovni jednotlivých správání) samostatné a dali sa jednoducho meniť a nahrádzať. Marián Lekavý uviedol svoju predstavu, že si to predstavuje podobne, ako pri 3D futbale, kde je celé správanie hráča rozbité do relatívne malých modulov. Ďalšia časť stretnutia bola zameraná na stanovenie úloh a dlhodobějších cieľov na nadchádzajúce obdobie.

Záver

1. Vedúcim bol definovaný zámer projektu.
2. Boli definované 2 tímy, z ktorých by bolo dobre podľa vedúceho vychádzať (Loptoši a možno tiež FIITBA).
3. Boli definované niektoré role a k nim prislúchajúce zodpovednosti (vedúci tímu, manažér kvality, manažér dokumentácie).

Revízia

Doposiaľ neboli definované nijaké predošlé úlohy.

Nové úlohy

| ID | Zodpovedná osoba | Úloha | Dátum zadania | Termín ukončenia | Skutočné ukončenie | Stav |
|-----------|-------------------------|--|----------------------|-------------------------|---------------------------|-------------|
| 1.1 | Všetci | | 12.10.2007 | 19.10.2007 | 13.10.2007 | vykonaná |
| 1.2 | Kasan | Podrobne si preštudovať stránku k projektu a svoje povinnosti. Pozrieť si dokumentáciu k 3D tímu 6th Sense, hlavne s ohľadom na architektúru hráča. | 12.10.2007 | 19.10.2007 | 18.10.2007 | vykonaná |
| 1.3 | Sekereš | Nájsť na internete nejaké zaujímavé architektúry autonómnych agentov a neobmedzovať sa pritom len na futbalistov. | 12.10.2007 | 19.10.2007 | 18.10.2007 | vykonaná |
| 1.4 | Všetci | Pozrieť si v správe niektorého z predchádzajúcich tímov, ako celá simulácia futbalu funguje. | 12.10.2007 | 19.10.2007 | 16.10.2007 | vykonaná |
| 1.5 | Všetci | Pozrieť si dokumentáciu vlaňajšieho tímu Loptoši. | 12.10.2007 | 19.10.2007 | 17.10.2007 | vykonaná |
| 1.6 | Nepšinský, Kováč | Pozrieť si dokumentáciu vlaňajšieho tímu FIITBA. | 12.10.2007 | 19.10.2007 | 18.10.2007 | vykonaná |
| 1.7 | Všetci | Napísať krátku správu o úlohách 1.2 až 1.6, ktorá sa stane súčasťou finálnej dokumentácie. | 12.10.2007 | 19.10.2007 | 18.10.2007 | vykonaná |

Zápis zo stretnutia č. 2

Číslo stretnutia: 2

Dátum: 19.10.2007 13:00

Miesto: Softvérové štúdio (D07b)

Zapisovateľ: Michal Kasan

Overovateľ: Peter Kajsa

Prítomní:

Pedagogický vedúci: Ing. Marián Lekavý

Členovia tímu: Bc. Michal Kasan, Bc. Peter Kajsa, Bc. Dávid Kováč, Bc. Martin Nepšinský, Bc. Branislav Kontúr, Bc. Tibor Sekereš

Téma stretnutia

Boli nám podrobnejšie opísané hlavné ciele projektu a taktiež jeho podstata. Marián Lekavý, vedúci nášho tímu, nám objasnil hlavné princípy fungovania simulovaného futbalu. Boli nám sčasti opísané programy ako monitor, klient, server so zameraním na funkcie týchto častí. Vedúci nám opísal úlohu trénera počas hry a tiež schopnosti jednotlivých hráčov. Definovali sme si pevný termín stretnutí na nadchádzajúce obdobie. Prevažná časť stretnutia bola zameraná na stanovenie úloh a dlhodobějších cieľov na nadchádzajúce obdobie.

Záver

1. Vytvorili sme si skupinový mail, ktorý budeme používať na komunikáciu medzi sebou a vedúcim tímu (TPteam06@googlegroups.com).
2. Zvolili sme si Loptošov ako tím, z ktorého budeme vychádzať a ktorého zdrojový kód budeme opravovať.
3. Z možných alternatív sme si zvolili Teamwork ako softvér na manažovanie projektu.
4. Boli definované niektoré role a k nim prislúchajúce zodpovednosti (vedúci tímu, manažér kvality, manažér plánovania, manažér podporných prostriedkov, správca webu, manažér dokumentácie).

Revízia zadaných úloh

Všetky krátkodobé úlohy boli splnené. Nijaké dlhodobé úlohy neboli zadané.

Nové úlohy

| ID | Zodpovedná osoba | Úloha | Dátum zadania | Termín ukončenia | Skutočné ukončenie | Stav |
|------|--------------------------------|--|---------------|------------------|--------------------|--------------|
| 2.1 | Všetci | Zvoliť si vedúceho. Jeho úloha je rozdeľovať úlohy a má zodpovednosť za ich plnenie. | 19.10.2007 | 25.10.2007 | 19.10.2007 | vykonaná |
| 2.2 | Všetci | Zvoliť si dokumentátora – Jeho úloha je „naháňať“, kontrolovať, dávať dokopy a kontrolovať dokumentáciu. | 19.10.2007 | 25.10.2007 | 19.10.2007 | vykonaná |
| 2.3 | Všetci | Zvoliť správcu stránky (je zodpovedný za stránku) a tiež zapisovateľa. | 19.10.2007 | 25.10.2007 | 19.10.2007 | vykonaná |
| 2.4 | Všetci | Opraviť štruktúru hráča s cieľom zabezpečiť modulárnosť (cez rozhranie triedy), aby sa dal ďalej rozširovať. Opraviť chyby v kóde. | 19.10.2007 | dlhodobá | | rozpracovaná |
| 2.5 | Všetci | Dodržať multiplatformovosť (server je exe bin súbor, ktorý sa spúšťa). | 19.10.2007 | dlhodobá | | rozpracovaná |
| 2.6 | Všetci | Urobiť pohľad na svet. | 19.10.2007 | dlhodobá | | rozpracovaná |
| 2.7 | Všetci | Urobiť refaktorizáciu tried. | 19.10.2007 | dlhodobá | | rozpracovaná |
| 2.8 | Kontúr, Kováč, Nepšínský | Identifikovať správania na určitej úrovni abstrakcie v zdrojových súboroch. | 19.10.2007 | 15.11.2007 | 13.11.2007 | vykonaná |
| 2.9 | Kajsa, Kasan, Sekereš | Prejsť strom dokumentácií a identifikovať správania. | 19.10.2007 | 15.11.2007 | 14.11.2007 | vykonaná |
| 2.10 | Kontúr | Zvoliť si vhodný softvér na zápis úloh a manažment projektu.. | 19.10.2007 | 25.10.2007 | 22.10.2007 | vykonaná |
| 2.11 | Všetci | Stiahnuť a odsimulovať zápas tímu Loptoši. | 19.10.2007 | 25.10.2007 | 25.10.2007 | vykonaná |
| 2.12 | Nepšínský | Vytvoriť skupinový mail. | 19.10.2007 | 25.10.2007 | 19.10.2007 | vykonaná |

Zápis zo stretnutia č. 3

Číslo stretnutia: 3

Dátum: 25.10.2007 13:00

Miesto: Softvérové štúdio (D07b)

Zapisovateľ: Michal Kasan

Overovateľ: Peter Kajsa

Prítomní:

Pedagogický vedúci: Ing. Marián Lekavý

Členovia tímu: Bc. Michal Kasan, Bc. Peter Kajsa, Bc. Dávid Kováč, Bc. Martin Nepšinský, Bc. Branislav Kontúr, Bc. Tibor Sekereš

Téma stretnutia

Na tomto stretnutí sme sa venovali častiam analýza, špecifikácia a hrubý návrh, pričom sa zadefinovalo, čo ktorá časť má obsahovať. Za najdôležitejšie bolo definované analyzovať správania hráča so zameraním na jeho architektúru. Tieto správania nachádzať tak v dokumentácii, ako aj v zdrojovom kóde. Podľa odporúčania vedúceho bolo rozhodnuté modularizovať nášho hráča podľa architektúry hráča tímu Hazard (z neho neskôr vychádzal tím 6th Sence). Novou dlhodobou úlohou na nadchádzajúce obdobie sa stalo rozbitie jednotlivých správání hráča, ktorého chceme modularizovať.

Záver

1. Bolo presne definované, čo je potrebné odovzdať do budúceho týždňa a taktiež o 2 týždne.
2. Boli definované termíny odovzdania jednotlivých častí dokumentu.
3. Definitívne sme sa dohodli na roliach a k nim prislúchajúcim zodpovednostiam na nadchádzajúce obdobie.

Revízia zadaných úloh

Všetky krátkodobé úlohy boli splnené. Na dlhodobých úlohách sa intenzívne pracuje.

Nové úlohy

| ID | Zodpovedná osoba | Úloha | Dátum zadania | Termín ukončenia | Skutočné ukončenie | Stav |
|-----------|-------------------------|--|----------------------|-------------------------|---------------------------|-------------|
| 3.1 | Nepšinský, Kováč | Nájsť vhodný softvér na zistenie metrík projektu (kvantitatívne stanoví kvalitu softvéru). | 26.10.2007 | 2.11.2007 | 31.10.2007 | vykonaná |
| 3.2 | Kasan | Vytvoriť správu zo stretnutí (zápis zo stretnutia) aj z predošlých stretnutí. | 26.10.2007 | 2. 11.2007 | 27.10.2007 | vykonaná |
| 3.3 | Všetci | Kompletne vypracovať časť analýza existujúcich hráčov fakulty. | 26.10.2007 | 2. 11.2007 | 4.11.2007 | vykonaná |
| 3.4 | Všetci | Získať dokumentáciu a zdrojové súbory tímu FIITBA od vedúceho a analyzovať architektúru hráča. | 26.10.2007 | 2. 11.2007 | 30.10.2007 | vykonaná |

Zápis zo stretnutia č. 4

Číslo stretnutia: 4

Dátum: 8.11.2007

Miesto: Softvérové štúdio (D07b)

Zapisovateľ: Michal Kasan

Overovateľ: Peter Kajsa

Prítomní:

Pedagogický vedúci: Ing. Marián Lekavý

Členovia tímu: Bc. Michal Kasan, Bc. Peter Kajsa, Bc. Dávid Kováč, Bc. Martin Nepšinský, Bc. Branislav Kontúr, Bc. Tibor Sekereš

Téma stretnutia

Na tomto stretnutí sme sa venovali častiam analýza, špecifikácia, ale predovšetkým hrubý návrh. Diskutovali sa predovšetkým správanie hráča, ďalej možná a plánovaná spolupráca modulov správania, hľadanie jednotlivých správání v zdrojovom kóde a tiež v dokumentácii a potreba explicitného vyjadrenia vzťahu medzi modulmi použitím stereotypu <<uses>>. Za najdôležitejšie bolo dodefinovanie dlhodobej úlohy a to rozbitie jednotlivých správání hráča, ktorého chceme modularizovať.

Záver

1. Bolo presne definované, čo je potrebné odovzdať do budúceho týždňa a čo má byť obsahom jednotlivých dokumentov.
2. Všetci členovia tímu jednotne pochopili zámer vedúceho a taktiež motív, a teda prečo je dôležité práve to, čo budeme robiť.
3. Bol stanovený postup hľadania jednotlivých správání a to začať od tých elementárnych správání poskytovaných serverom(zdola nahor).
4. Bola došpecifikovaná dlhodobá úloha "Urobiť pohľad na svet" na vytiahnutie „trvalejších“ informácií „von“, a teda ich umiestnenie do samostatného modulu.

Revízia zadaných úloh

Všetky krátkodobé úlohy boli splnené. Na dlhodobých úlohách sa intenzívne pracuje.

Nové úlohy

| ID | Zodpovedná osoba | Úloha | Dátum zadania | Termín ukončenia | Skutočné ukončenie | Stav |
|-----------|-------------------------|---|----------------------|-------------------------|---------------------------|-------------|
| 4.1 | Všetci | Každý si pozrie, čo robí server. | 8.11.2007 | 15.11.2007 | 15.11.2007 | vykonaná |
| 4.2 | Kasan | Vytvoriť správu zo stretnutí (zápis zo stretnutia). | 8.11.2007 | 15. 11.2007 | 9.11.2007 | vykonaná |
| 4.3 | Kasan | Vytvoriť preberací protokol. | 8.11.2007 | 15. 11.2007 | 12.11.2007 | vykonaná |
| 4.4 | Všetci | Skompletizovať analýzu, špecifikáciu a hrubý návrh do uceleného dokumentu. | 8.11.2007 | 15. 11.2007 | 14.11.2007 | vykonaná |
| 4.5 | Sekereš | Identifikovať elementárne správanie poskytované serverom a zapracovať to do jednotlivých častí dokumentu. | 8.11.2007 | 15. 11.2007 | 11.11.2007 | vykonaná |