## 1. Úprava logovania

* Odstranenie loggera v baliku sk.fiit.jim.log
* Vytvotenie noveho jednotneho loggera, ktory pouziva ako zaklad standardny logger z JavaAPI obaleny triedou JLog
* Logger umoznuje logovat na konzolu prostrednictvom triedy JimConsoleFormatter alebo do HTML tabulky prostrednictvom triedy JimHtmlFormatter
* Rozsirenie loggera o nove typy v súbore LogType, ktore je mozne logovat pouzitim metody JLog.addLogType(LogType l), pripadne je mozne logovat vsetky typy pouzitim metody JLog.addAllLogTypes():

LogType . INIT

LogType . AGENT\_MODEL

LogType . WORLD\_MODEL

LogType . INCOMING\_MESSAGE

LogType . OUTCOMING\_MESSAGE

LogType . LOW\_SKILL 50

LogType . HIGH\_SKILL

LogType . GUI

LogType . TACTIC

## 2. Odstranenie zakomentovanych casti kodu

* Vymazane cele triedy:

OldReplanWindow z balicka sk.fiit.jim.gui – bola nahradena novou triedou

TestJim z balicka sk.fiit.jim.tests – nepouzivala sa a nebola funkcna

KickSkills z balicka sk.fiit.jim.agen.kickskill.kick – prazdna a nepouzivala sa

Prophecy z balicka sk.fiit.jim.agent.models.prediction + funkcia calculateBallPosition()

* Vytvoreny novy balik sk.fiit.jim.decision.tactic.oldgoalie pre uchovanie starého brankára, do ktorého boli presunuté nasledujúce triedy:

Goalie - presunuté z sk.fiit.jim.decision.tactic

GoalieTest - presunuté z sk.fiit.jim.tests

GoalieTestCase - presunuté z sk.fiit.jim.tests

GoalieTestCaseTest - presunuté z sk.fiit.jim.tests

## 3. Odstranenie balika sk.fiit.jim.garbage

* Zdruzoval nepotrebne baliky, ktore sa po prepisani casti kodu agenta z Ruby do Javy dalej nepouzivali

## 4. Úprava pôvodného zdrojového kódu podľa konvencií písania zdrojového kódu

* vytvorená metodika písania zdrojového kódu 31 pomocou ktorej bol analyzovaný pôvodný zdrojový kód
* doplnene dokumentačné komentáre
* telá podmienených príkazov a cyklov boli uvádzane do blokov
* kód bol správne formátovaný, čim kód dosiahol lepšiu prehľadnosť a tým sa zvýšila aj jeho kvalita

## 5. Odstránenie nepoužívaných highskillov

* nepoužívané highskilly boli odstránené z projektu a zdrojové súbory boli presunuté do priečinka OLD FILES mimo hlavný projekt agenta

## 6. Oddelenie testov od zdrojových kódov

* testovacie triedy sa presunuli do nového zdrojového priečinka určeného pre testy
* V projekte JIM sa vytvoril nový zdrojový priečinok Test
* Následne sa presunuli všetky testovacie triedy JUnit testov do príslušných balíkov zdrojového priečinka Test

## 7. Implementácia Zero moment point

* Hlavná funkcia pre výpočet Zero moment point bola zakomponovaná do triedy AgentModel a ide o triedu private void updateZeroMomentPoint(), v ktorej sa počíta bod, v ktorom by mal byt hráč stabilný
* ďalšie pridané metódy do AgentModel boli private void updateCenterOfMass(), private void updateBodyPartsPositions2(), private void updateFeetForce(ParsedData data) a public boolean isOnGround()
* V triede Vector3D pribudli metódy public Vector3D normalize(), public Vector3D rotateOver(Vector3D axis, double angleInRad) a public Vector3D atten(), ktoré sa využívajú pri práci s vektormi v Zero moment point

## 8. Zámena triedy Vector3 za Vector3D

* V projekte Jim a v testovacom frameworku v projekte TestFramework sa nachádzalo približne 300 použití triedy Vector3
* Všetky použitia boli úspešne nahradené za Vector3D a trieda Vector3 bola odstránená z projektu
* Z triedy Vector3D bola na záver odstránená konverzná metóda z triedy Vector3

## 9. Pridávanie hrá£ov do testovacieho frameworku

* možnosť pomocou testovacieho frameworku pridávať hráčov v rámci dvoch rôznych tímov

## 10. Refaktorovanie projektu RoboCupLibrary

* Všetky členské premenné, ktoré bolo možné označiť ako private sme takto označili. V niektorých triedach boli vytvorené aj príslušné gettery a settery
* Triedu Vector2 sme premenovali na Point2D a vložili sme do nej metódu pre výpočet vzdialenosti dvoch bodov, ktorú sme zase odstránili z triedy MEC
* V rámci refaktoringu boli zdokumentované všetky triedy a metódy a zdrojový kód bol správne naformátovaný podla metodiky pre písanie zdrojového kódu

## 11. Príprava taktík pre RoboCup turnaj

* Boli upravene už existujúce turnajové taktiky tak, aby zodpovedali pravidlám. Tie taktiky, ktoré neexistovali sme vhodne doplnili:

GetUpFromStomach

GetUpFromBacks

KickDistance

Turn180

StabilityWalkTactic

FastWalkTactic

KickAccuracy

KickOnGoal a KickOnXY

## 12. Analýza chybných JUnit testov

* zanalyzovane chyby v testoch a situáciách, pri ktorých testy dávajú negatívy výsledok. Následne bol podla výsledkov analýzy vyvodený záver a vykonaná zmena v kóde

## 13. Tvorba HighSkills a zakomponovanie Zero moment point

* Postupnosť vykonávania HighSkillu a jeho implementácia
* Stabilizácia pomocou Zero moment point

## 14. Otáčanie hráča

* Otáčanie hráča implementovane do highskillu TurnToVector