**Model sveta**

Model sveta zhromažďuje informácie, ktoré sú potrebné pri vývoji a testovaní agenta. Model sveta možno rozdeliť na:

* Súčasný stav agenta: pozície kĺbov, globálna poloha a rotácie vzhľadom na globálnu os.
* Súčasný stav okolia: stav hry, mód hry, hrací čas, simulačný čas a verzia servera.
* Stav sveta: pozície ostatných hráčov a lopty.

# Model sveta v kóde

Model sveta je v kóde zahrnutý pod komponentou Models. Je rozdelený do nasledujúcich častí:

* AgentModel –stav agenta
* EnvironmentModel – stav okolia
* WorldModel - stav sveta

## Triedy

### AgentModel.java

Trieda, ktorá obsahuje určovanie súčasný stav agenta a to z pohľadu jeho pozície kĺbov, globálnej polohy a rotácie vzhľadom na globálnu os. Ďalej určuje pozíciu hráča, teda na ktorej strane hráč je (člen tímu, súper) metódu na výpočet pohybu vektora okolo vektorov a osí. Výpočty sú vykonávané podľa teórie vektorových pohyb nájdených na internete. Táto trieda spolupracuje s akcelometrom a práve pomocou stavu agenta dokáže identifikovať, či agent stojí alebo leží na zemi. Obsahuje GETY na získanie informácii o tom či hráč stojí, je na zemi, leží na chrbte, leží na bruchu, ďalej o natočení a pozícii samotného hráča. Obsahuje SETY na nastavenie pozície a rotácie kĺbov.

### AgentPositionCalculator.java

Trieda, ktorá vypočítava aproximáciu aktuálnej polohy hráča na základe vlajok, ktoré vidí. Na základe zistení vytvára logy. Predpokladá, že rotácie hráča sú vypočítané správne. Výpočet je vykonávaný normalizáciou a odpočtom od pozície zástavy. Keď vnímaná relatívna pozícia je pridaná k známej pozícii vlajky, je možné vypočítať správnu aproximáciu pozície hráča.

### AgentRotationCalculator.java

Trieda, ktorá vypočítava natočenia hráča na základe vlajok. Dokáže určiť natočenie agenta z 3 videných vlajok. Určí, ktoré dve z troch použije na vytvorenie osi Y , a ktorú doplní bod na vytvorenie osi Z, kolmej na os Y.

### DynamicObject.java

Trieda určuje polohu dynamických objektov, zvyčajne lopty.

### EnvironmentModel.java

Trieda, v ktorej sa ukladajú statické hodnoty o stave okolia, ako sú stav hry, mód hry, hrací čas, simulačný čas a verzia servera.

### FixedObjects.java

Trieda obsahujúca vlajky videné perceptorom SEE robota. Každá vlajka uchováva informáciu o svojej pozícii. Určuje sa tak veľkosť ihriska.

### KalmanAdjuster.java

Trieda, ktorá nastavuje vnímané súradnice lopty a vlajky na ihrisku použitím kalmanovho filtra, čím sa znižuje chybovosť pre ďalšie výpočty.

### Player.java

Trieda, ktorá definuje druhého hráča a jeho rolu v hre (druhého hráča, ktorého vidí).

### TacticalInfo.java

Trieda, ktorá obsahuje informácie o herných situáciách (útočíme, bránime...) a tiež metódy pre vytvorenie hernej formácie.

### WorldModel.java

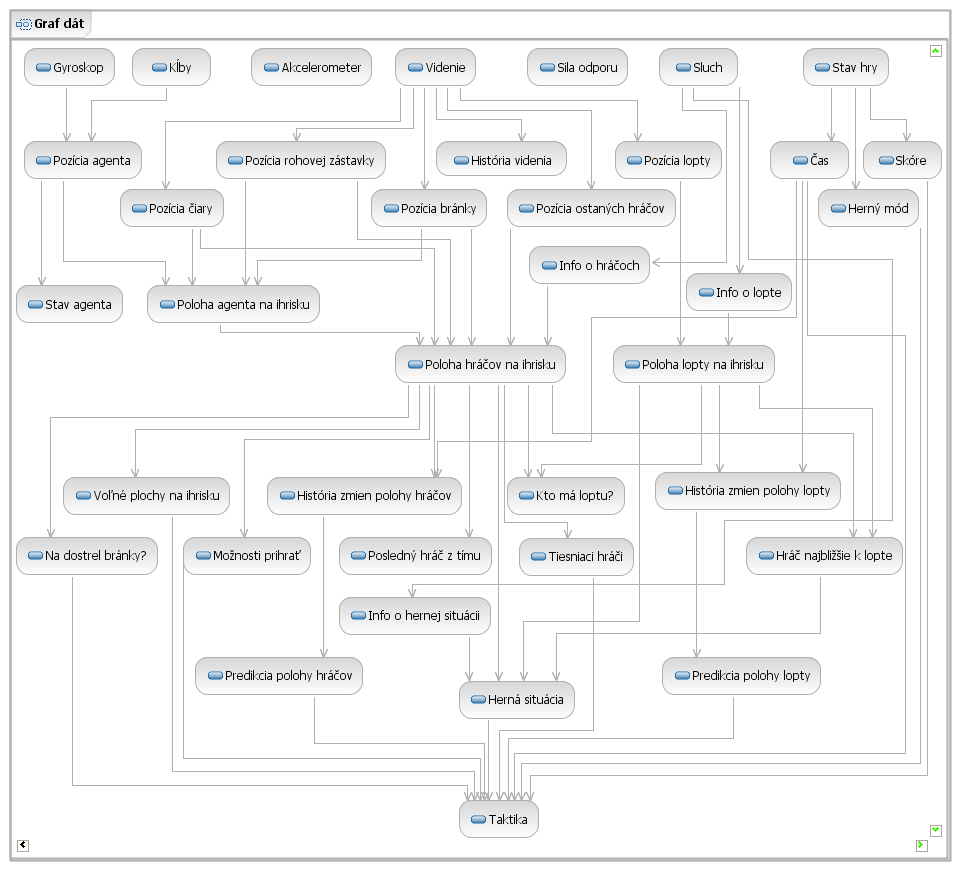
Trieda, v ktorej sa ukladajú objekty na ihrisku, teda ich pozícia, rýchlosť a priemernú pozorovanú rýchlosť. Počíta pozície ostatných hráčov a lopty.

# Dáta v modeli sveta

V modeli sveta figuruje viacero dát, na základe ktorých samotný model sveta funguje. Jednotlivé dáta boli spracované do grafu dát a vysvetlené.

## Graf dát

Graf dát predstavuje informácie, ktoré dokáže agent o svete získať. Informácie sú na rôznych stupňoch abstrakcie. Najnižšiu vrstvu predstavujú tie, ktoré prichádzajú zo servera a predstavujú dáta, ktoré agent dostáva zo snímačov. V grafe sú to uzly, do ktorých nevchádza žiadna hrana. Model grafu dát je zobrazený na obrázku 1.



Obrázok 1: Graf dát

## Dáta

Dáta ktoré figurujú v modeli sveta sú vymenované a popísané nižšie.

### Gyroskop

Údaje z perceptora GyroRate v tvare:

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

### Kĺby

Údaje z perceptorov typu HingeJoint a UniversalJoint v tvare:

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

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

### Sila odporu

Údaje z perceptorov ForceResistance v tvare:

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

### Sluch

Údaje z perceptorov Hear v tvare:

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

### Akcelerometer

Údaje z perceptorov Accelerometer v tvare:

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

### Videnie

Údaje z perceptora Vision v tvare:

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

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

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

### Stav hry

Údaje z perceptora GameState v tvare:

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

### Pozícia bránky

Relatívna pozícia bránky vzhľadom na agenta.

### Pozícia rohovej zástavy

Relatívna pozícia rohovej zástavy vzhľadom na agenta.

### Pozícia lopty

Relatívna pozícia lopty vzhľadom na agenta

### Pozícia ostatných hráčov

Relatívna pozícia hráčov vzhľadom na agenta

### Pozícia čiar

Relatívna pozícia čiar vzhľadom na agenta

### Herný mód

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

### Poloha agenta na ihrisku

Poloha agenta na ihrisku. Absolútna poloha.

### Pozícia agenta

Pozícia agenta v akej sa nachádza. To je poloha ťažiska a jednotlivých končatín.

### Stav agenta

Polohe, ktorej je agent a čo sa s ním deje (Stojí, Leží, Padá).

### Poloha lopty na ihrisku

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

### Poloha ostatných hráčov na ihrisku

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

### Herná situácia

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

### Hráč najbližšie k lopte

Hráč, ktorý je aktuálne najbližšie k lopte.

### Voľné plochy na ihrisku

Plochy na ihrisku, kde sa nenachádzajú protihráči.

### Posledný hráč z tímu

Hráč, ktorý predstavuje brankára.

### Čas

Aktuálny čas.

### Skóre

Aktuálny stav zápasu.

### História zmien polohy lopty

Polohy lopty za posledných x sledovaných momentov.

### Predikcia polohy lopty

Predpokladaná poloha lopty v nasledujúcom momente.

### História zmien polohy hráčov

Polohy hráčov za posledných x sledovaných momentov.

### Predikcia polohy hráčov

Predpokladaná poloha hráčov v nasledujúcom momente.

### Taktika

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

### Kto má loptu?

Ktorý hráč má loptu.

### Možnosti prihrávky

Všetky aktuálne možnosti na prihrávku spoluhráčovi.

### Tiesniaci hráči

Všetci hráči v nebezpečnej vzdialenosti od agenta.

### Na dostrel bránky?

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

Údaje o type a formáte prichádzajúcich dát zo servera sú zo stránky: http://simspark.sourceforge.net/wiki/index.php/Effectors, tiež je tam možné nájsť detailnejšie informácie k jednotlivým efektorom a popis k formátu dát.