

## Inštalácia

1. V prvom rade som sa pokúšal rozbehať projekt na svojom notebooku, na ktorom beží operačný systém windows 10, inštaláciu som robil podľa návodu, no projekt sa mi nedal rozbehať. Po konzultácii s kolegami sme dospeli k záveru, že by bola potrebná reinštalácia windowsu.
2. V druhom rade som skúšal inštaláciu na macOSx prostredníctvom virtualboxu a windowsu 10. V tomto prípade som taktiež postupoval podľa návodu na stránke no nebolo možné získať licenciu na windows 10 prostredníctvom konta na študenta cez dreamspark tak som si stiahol iba trial verziu na 90 dni, ktorá by mala byť postačujúca na dokončenie projektu. No vo virtualboxe sa mi projekt na windows 10 taktiež nepodarilo rozbehať.
3. V poslednom rade som skúšal inštaláciu prostredníctvom bootcampu na macu, kde som si nahral windows 10 a následne som postupoval podľa návodu na stránke, kedy sa mi projekt konečne podarilo rozbehať.

## Analýza projektu / wiki

Z analýzy RobocupTP wiki. Som zistil základné informácie o robotoch.

**Aktuálne pohyby** robota sú, chôdza, kopnutie, úkroky, otáčania, vstávanie a sadanie, pády.

**Hráč/robot** sa skladá z 22 kĺbov 6 v každej nohe, 4 v každej ruke a 2 v krku.

**High Skill**

**Low Skill**

Funkcionalita low skillov, high skillov, taktík a stratégií bola vysvetlená na tímovom stretnutí členom tímu Ernest Loureiro.

## Perceptory

Na vnímanie sveta v RoboCup 3D slúžia perceptory. Server zasiela agentovi informácie o samotnom agentovi, videných objektoch a podobne.

**GyroRate** slúži na opis orientácie tela robota vzhľadom k jednotlivým osiam. Tento perceptor je umiestnený v hornej časti robota Nao.

**HingeJoint** určuje o koľko stupňov je ktorý kĺb natočený.

**ForceResistance perceptor** slúži na oznámenie o pôsoiacej sile na časť tela a jej vektore. Správa obsahuje dvojce súradnice, jedny určujú bod, na ktorý sila pôsobí a druhé súradnice určujú vektor tejto sily

**Accelerometer perceptor** umiestnený v hornej časti tela počíta zrýchlenie.

**Vision perceptor** dostáva informácie o objektoch videných agentom (ostatní agenti, lopta, statické body ihriska). Zorné pole robota Nao je 120°.

**GameState** vysiela niekoľko základných informácií o aktuálnom stave hry.

**HearPerceptor** – priama komunikácia medzi robotmi nie je povolená, ale môžu si vymieňať správy cez simulačný server. Na túto komunikáciu slúži tento perceptor.

## Efektory

**Create** – v momente ako je agent pripojený na server nie je viditeľný a nemôže zasiahnuť do simulácie. Pomocou Create hráč odovzdá serveru cestu ku konfiguračnému súboru, ktorý definuje fyzickú reprezentáciu agenta a množinu jeho perceptorov a efektorov.

**HingeJoint** efektor dávajúci príkaz na ohnutie kĺbu.

**Beam efektor** určuje umiestnenie agenta na hracej ploche pred začatím polčasu

**Say efektor** slúži na odoslanie správy pre komunikáciu medzi hráčmi.

**Init** priradí hráča k tímu a prideli mu číslo. Ak agent posiela ako číslo 0, server priradí hráčovi najbližšie voľné číslo automaticky

## Analýza modelu sveta

Model sveta je obsiahnutý v komponente Models. Kde sa nachádzajú 3 časti, a to AgentModel, EnvironmentModel a WorldModel. V triede AgentModel sa počítajú pozície a rotácie kĺbov, v triede WorldModel sa počítajú pozície ostatných hráčov a lopty a v triede EnvironmentModel sa ukladá mód hry, čas a verzia servera.

## Analýza matiky a logiky

V tejto časti sú popísané výpočty jednotlivých tried.

## Analýza výstupov z gyroskopu a akcelerometra

V tejto časti je vykonaná analýza dát z jednotlivých receptorov.

## Jednoduché pohyby

Táto podkapitola sa zameriava na analýzu jednoduchých pohybov. Od vstávania z brucha až po kop do lopty.

## Komunikácia agenta so serverom

Celá hra prebieha ako interakcia medzi agentom a serverom. Server má informácie o aktuálnom stave hry, tj. Rozloženie hráčov, čas, poloha lopty, ... . Tento stav sa dá graficky zobrazit pomocou aplikácie rcssmonitor3d. Server poskytuje minimálne dva typy robotov – „Nao“ a „Soccerbot“. Nao predstavuje virtuálnu reprezentáciu rovnomenného robota a je používaný aj v projekte Robocup na FIIT.

## High Skill

XML súbory s low skillmi a fázami musia pre použitie v high skilloch okrem vyššie uvedených požiadaviek spĺňať nasledovné: každý musí mať aspoň jednu fázu s isFinal=true

Je tu taktiež opísaný priebeh vykonávania high skillu a sú tu aj opísané existujúce high skillly

## Low Skill

Jednotlivé nižšie pohyby agenta sú opísané prostredníctvom XML súborov (nižšie pohyby sú napríklad kopnutie do lopty, postavenie sa, otočenie hlavou a podobne). Každý pohyb sa skladá z viacerých fáz, pričom každá fáza predstavuje určitú zmenu natočenia určitých kĺbov za určitý čas. V tejto kapitole je taktiež popísaná štruktúra xml súborov.

## Vyhodnocovanie polohy agenta

Výpočet polohy agenta prebieha v triede AgentPositionCalculator. Výpočet polohy pozostáva z nasledujúcich krokov:

1. Prijatie správy zo servera
2. Sparovanie informácií
3. Vyhodnotenie polohy na základe jednej čiary
4. Mapovani čiar ak agent vidí iba čiary
5. Mapovani čiar ak agent vidí aspoň jeden kontrolny bod
6. Výpočet polohy agenta
7. Aktualizácia polohy
8. Aktualizácia histórie agenta

Všetky tieto kroky sú v kapitole podrobne vysvetlené.

## Správy zo servera

Správu zo servera agent dostáva približne každých 0.02s, pričom každá tretia takáto správa obsahuje informácie o tom, čo agent vidí (podľa simspark wiki by mala každá druhá správa obsahovať informácie o tom, čo agent počuje).

## Vytvorenie pohybu v editore

V tejto kapitole je opísaný celý proces vytvárania pohybu v editore.

## Vzory Čiar na Ihrisku

V tejto sekcii je opísaná trieda LinePatternRecognition a aj všetky jej funkcie.

## Analýza metodík

V rámci tohto šprintu som si prešiel aj jednotlivé metodiky. A to konkrétne:

- **Metodika konvencie kódu.** – metodika sa zaoberá tým ako by mal byť správne písaný kód, ako písať komentáre a ako formátovať kód a ako robiť deklarácie.
- **Metodika kontrolovania kódu.** – metodika sa zaoberá tým ako kontrolovať kód a opisuje celý postup od pull requestov až po opravenie kódu.
- **Metodika vykazovania prác v nástroji scrumdesk.** – metodika opisujúca celý proces vykazovania práce, definuje typy úloh
- **Metodika písania dokumentov.**
- **Metodika komunikácie.** – metodika opisuje spôsobom komunikácie medzi členmy tímu, definuje uložiská a aj nástroje v ktorých sa komunikuje.
- **Metodika rizík.**
- **Metodika testovania.**
- **Metodika commitovania a verziovania.**