Tietorakenteiden ja algoritmien harjoitustyö
Lari Saukkonen

Toteutusdokumentti

Ohjelman toteutus

Valitsin aiheekseni tekoälyn, koska annetuista aiheista se herätti eniten mielenkiintoa. Uskon myös että voin käyttää oppimaani myöhemmin konkreettisesti hyväksi. En ollut aiemmin perehtynyt aiheeseen ja se tuntui minulle hieman kaukaiselta. Ennakkokäsitykseni tekoälystä oli, että se on jotain todella vaikeaa ja lähes mahdotonta tehdä. Päätin kuitenkin yrittää koska aihe selvästi motivoi minua.

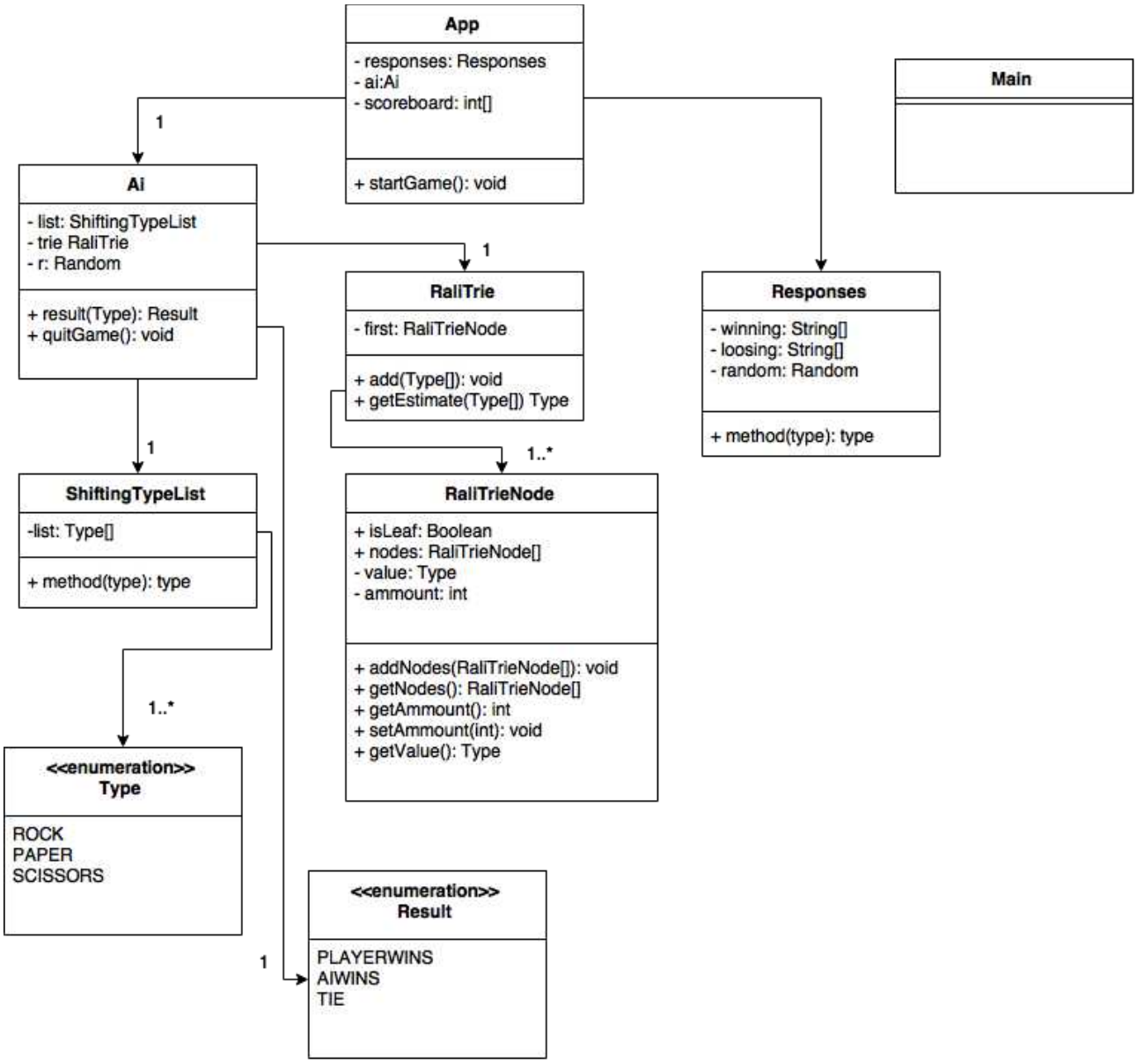
Ohjelman toimintaperiaate nojaa Trie-tietorakenteeseen joka pitää kirjaa edellisistä valituista kombinaatioista. Ohjelma suorittaa tallennuksia ja hakuja viiden sarjoissa mutta se voidaan nopeasti muokata mihin tahansa muuhun sarjapituuteen tai jopa vaihtelevaan sarjapituuteen. En kuitenkaan nähnyt tarpeelliseksi tehdä vaihtelevaa sarjapituutta, koska ohjelma serialisoi tietorakenteen joka käynnistyksen jälkeen ja näin oppii peleistä. Tällöin sarjatietoa on aina käytettävissä.

Laajensin ohjelman ottamaan huomioon myös tekoälyn edelliset siirrot valintojen tueksi. Näin tiedot ovat siis 5 + 5 (pelaaja , tekoäly) muodossa. Mielestäni muutos paransi oleellisesti tekoälyn pärjäämistä pelaajaa vastaan. Ohjelma on suunniteltu pääsiassa käytettäväksi yhtä ihmispelaajaa vastaan.

Aihe herätti minussa yleistä mielenkiintoa tekoälyä kohtaan ja selailinkin fuksiwikin kirjoista itselleni sopivan kirjan, jolla voisin lukea tekoälystä lisää ja samalla varustautua jo ennalta ”Johdatus tekoälyyn”-kurssia varten.

Ohjelmaa voisi vielä laajentaa esimerkiksi muodostamalla profiileja eri pelaajista ja käyttämällä eri Trie-rakennetta aina kun erilainen pelityyli tunnistetaan. Tällöin ohjelma pystyisi vastaamaan paremmin muuttuviin tilanteisiin ja mukautumaan aina jokaisen pelaajan pelityyliin sopivaksi.

Ohjelman rakenne



Aika- ja tilavaativuudet

Toiminnan kannalta olennaisimman osan Trie:n toiminta on nopeaa ja suoraaviivaista. Rekursiivisia haku- ja talletusmetodeja on tehostettu käyttämällä ehtolauseita ennen jokaista haaraa, eikä lista järjestä itseään uudelleen johtuen sen rakenteesta.

Haku ja lisäys tapahtuvat ajassa O(n) ja tilavaatimus on myös O(n). Puun koosta johtuen rekursiokaan ei aiheuta minkäänlaisia ongelmia suorituksen kannalta. Yritin myös selvittää, että kääntääkö JVM indeksoidut rekursiot loopeiksi kuten Scalassa mutten löytänyt vastausta.

Jatkokehitys

Ohjelma toimii moitteettomasti ja tekoälyn toiminta on mielestäni hyvällä tasolla. Tällä hetkellä ohjelma osaa varautua peliin yhtä pelaajaa vastaan ja muistaa siirrot pelikertojen yli. Sain myös valmiit tietorakenteet korvattua omillani.

Ohjelmaa voisi laajentaa tukemalla montaa pelaajaprofiilia ja tallentamalla jokaiselle oman RaliTrie:n edellisten siirtojen muistamiseksi. Ollakseen loistava, tekoälyn täytyisi myös osata analysoida tehtyä siirtoja ja pelitilannetta vanhojen siirtojen toimivuuden valossa, eli esimerkiksi muuttaa strategiaa mikäli tallennettu tieto ei toimisikaan pelin edetessä tai uudessa pelissä.