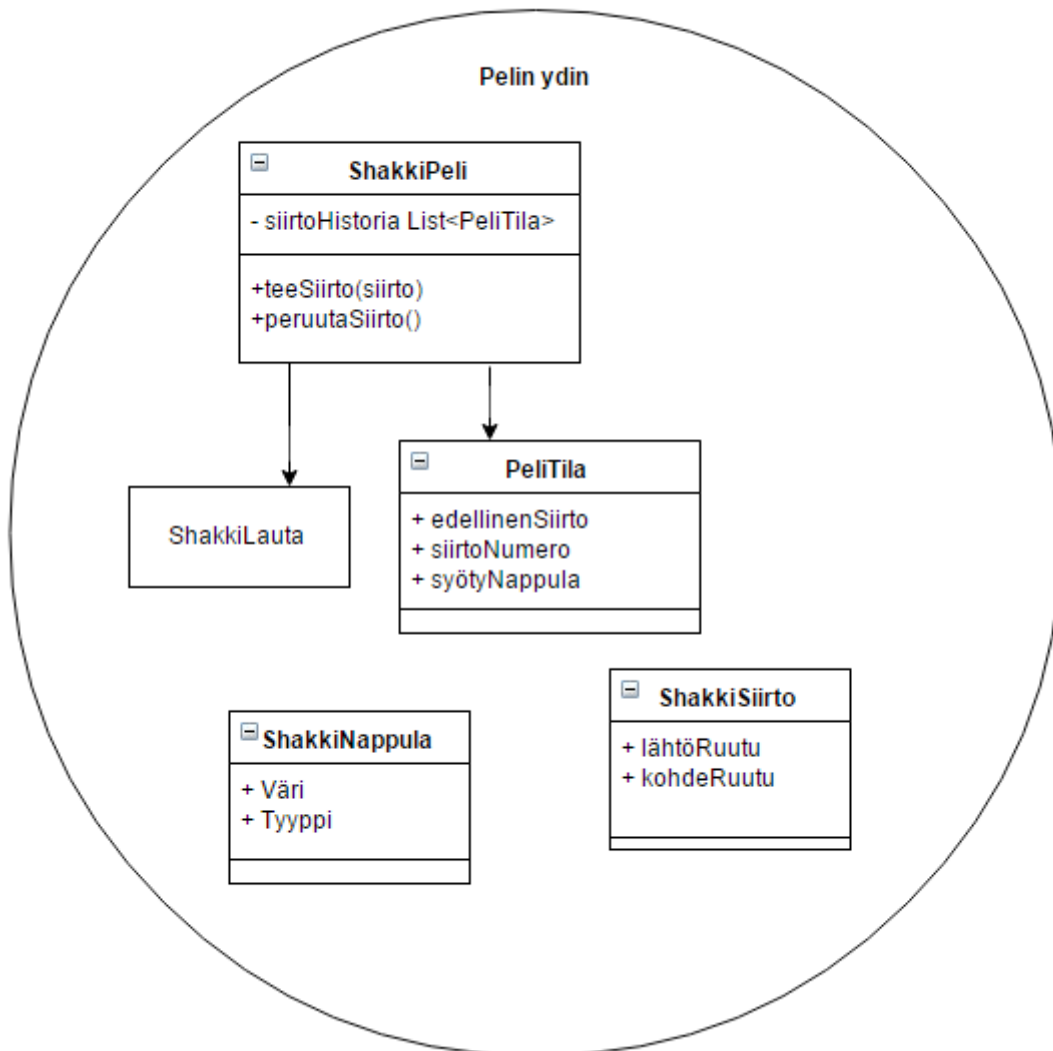


ShakkiTekoäly
tira harjoitustyö HY CS

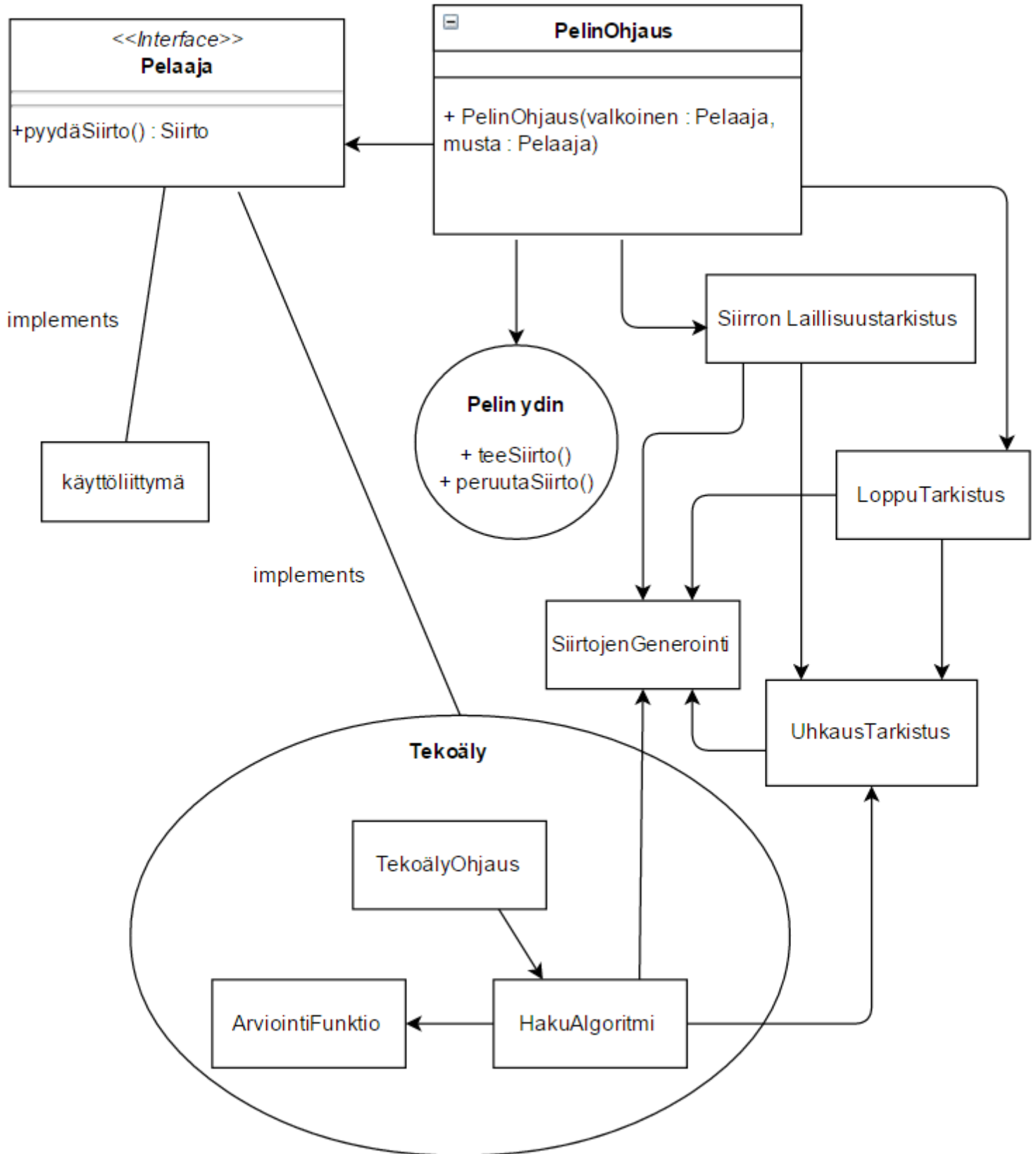
toteutusdokumentti
Markus Mustonen

Ohjelman rakenne

Pelin ydin:

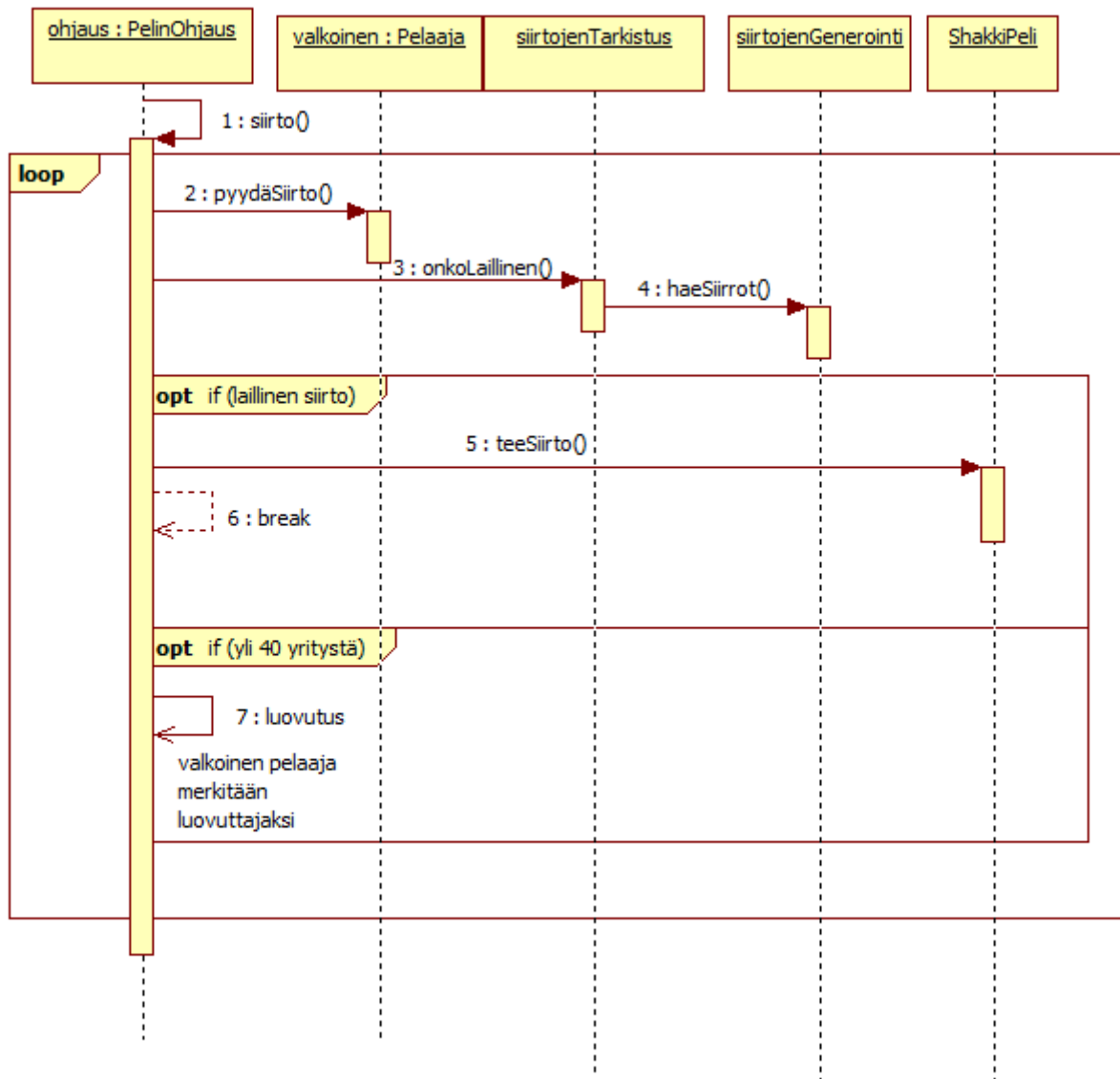


Kokonaisuus:
Nuolet kuvaavat riippuvuuksia.



Ohjelman toiminta:

Seuraava kaavio kuvaa PelinOhjauksen siirto-metodin kutsua, kun on valkoisen pelaajan vuoro. Pelaajalta pyydetään siirtoa niin kauan, kunnes vastauksena saadaan laillinen siirto.



Rakenne ja toiminta näkyvät tarkemmin javadoc-dokumentaatiosta.

Algoritmi ja suorituskyky

Minmax-algoritmin aikavaativuus on exponentiaalinen, esim. 30^n riippuen pelin asetelmasta. Alfabeta-karsinnalla saadaan huomattava parannus aikavaativuuteen, mutta se säilyy edelleen exponentiaalisena ks. testausdokumentti. Suorituskyvyn vuoksi hakualgoritmissa ei testata jokaisen siirron laillisuutta täsmällisesti vaan lailliset siirrot kumotaan sopivilla arvoilla ja pienillä tarkistuksilla. Suhteellisen tehokkaalla koneella haku käsittelee noin kolme miljoonaa siirtoa sekunnissa.

Parannettavaa

Algoritmia voisi vielä optimoida esimerkiksi tutkimalla siirtoja tietyssä järjestyksessä, jotta beeta-katkaisu tapahtuisi mahdollisimman ajoissa, varsinkin lähellä pelipuun juurta. Tätä varten voisi pitää kirjaa siirroista, jotka ovat aiheuttaneet katkaisun. Jos arviointifunktiona on simppeli materiaaliarviointi, niin jokaista peliasetelmaa ei tarvitsisi arvioida erikseen, vaan arvoa voisi muuttaa aina syövien siirtojen kohdalla. Lauta kannattaisi tallentaa oliotaulukon sijaan kokonaislukumuuttujina, joissa bitit vastaavat nappuloita. Näin siirtojengenerointi voisi tapahtua nopeasti binäärioperaatioilla. C++ olisi tehokkuuden puolesta parempi kieli tekoälyn koodaamiseen...