

(Pelin yleisrakenne löytyy dokumentointikansiosista. Siellä on myös ohjeet pelistä.)

Terminologiaa

Player 1: aloittava pelaaja, joka pelaa ruudun alaosassa

Player 2: toinen pelaaja, joka pelaa ruudun yläosassa

Player A: pelaaja kenen vuoro on

Player B: toinen pelaaja

AI

Kaikki pelin tekoälyt perivät abstraktin luokan AI, jolta löytyy metodi `playTurn()`. Tekoäly lisätään pelin Launcher-luokassa pelaajalle/pelaajille ja se välitetään Game:n ja UserInterfacen kautta logiikasta vastaavalle LogicHandler-luokalle, joka pelin alussa lisää tekoälyt pelaajille (Player) metodilla `setupAI()`. Jos pelaajalle on lisätty tekoäly, niin pelaajan vuoron alussa LogicHandler kutsuu AI:n `playTurn()`-metodia eli pelaa vuoron ja antaa vuoron takaisin toiselle pelaajalle (metodi `playAI` LogicHandlerissa). AI-luokalla on monia protected metodeita, joita moni eri AI hyödyntää.

SimpleAI

SimpleAI on tekoälyn toteutus, joka tekee kaikki päätökset perustuen satunnaisuuteen. Sen käyttämät `playTableRandomly` ja `playHandRandomly` -metodit on toteutettu AI-luokassa, jotta muutkin tekoälyt voivat hyödyntää niitä.

SimpleAI aloittaa vuoron hyökkäämällä pöydässä olevilla korteillaan satunnaisia kohteita. Sitten se pelaa kortteja kädestä pöytään siinä järjestyksessä missä ne on nostettu käteen (mikä tapahtuu Randomilla). Lopuksi se vielä hyökkää pöydässä olevilla Mounted-korteilla satunnaisia kohteita.

MediumAI

MediumAI on toinen tekoälyn toteutus, joka osaa jo jotain. Se eroaa SimpleAI:sta siten, että vuoron alussa se tarkistaa `checkLethal()`-metodilla voiko se voittaa tällä vuorolla ja se viimeistelee pelin jos mahdollista, ja pelatessaan kortteja pöytään se osaa sijoittaa ne paikkoihin, joka suosii optimaalista Guardianien sijoittamista suojaamaan muita, ja jos pöydässä on Guardian se pelaa muun Minionin sen viereen, ja jos pöydässä on muu Minion, se pelaa Guardianin sen viereen. Hyökätessään Minioneilla, se yrittää ensin hyökätä kohteita jotka se voi tappaa yhdellä hyökkäyksellä: yrittäen ensin kohteita joiden health on sama kuin hyökkääjän attack ja sitten kohteita joiden health on vähemmän.

MediumAI pelaa tavalla, joka muistuttaa huonoa ihmispelaajaa. Testaamalla sitä SimpleAI:ta vastaan se kuitenkin voittaa enemmän pelejä (voitto-häviö-ratio ~2.252).

AdvancedAI

AdvancedAI ottaa huomioon pöydässä olevat omat ja vastustajan Minionit, ja se tekee sen perusteella päätökset miten se pelaa. Se eroaa MediumAI:sta siinä, miten se pelaa Minionit kädestä pöytään. Riippuen siitä onko vastustajan pöydässä Minioneita, se kokeilee pelata eri tyyppisiä Minioneita järjestyksessä, joka on optimoitu testaamalla AI:ta AITesterillä.

Se antaa jo melko vaikean vastuksen. SimpleAI:ta vastaan se voittaa ratiolla ~3.702 ja MediumAI:ta vastaan ~3.058.

AITester

Tekoälyjä voidaan testata vastakkain vaihtamalla Launcherissa ehtolauseke true:ksi, jolloin peli ei käynnistä käyttöliittymää lainkaan vaan pelaa tekoälyillä toisiaan vastaan ja antaa molempien voitot palautteena. Tämä tehdään AITester-luokan avulla, jolla on metodi runTests(int amount). Parametrilla määritellään siis pelattavien pelien määrä. Toimiakseen molemmille pelaajille pitää olla annettu tekoälyt.

Puutteita ja parannusehdotuksia

Pelissä ei ole vielä menua, jonka kautta peli voitaisiin käynnistää ja asetuksia voitaisiin vaihtaa. Nyt pelin jar-tiedosto aloittaa suoraan pelin AdvancedAI:ta vastaan. Vaihtaakseen eri AI:den välillä tai muita asetuksia, se joudutaan tekemään Launcher-luokan koodissa.

Mouse hoverit on toteutettu vasta pelaajan omalle kädelle, ja sekin on hieman puutteellinen: hitaammilla tietokoneilla jää välillä monta korttia zoomatuksi samaan aikaan jos hiirtä liikuttaa nopeasti. Käden piirtäminen pöytään on myös hieman hölmösti toteutettu tällä hetkellä, ne vaihtavat paikkoja ruudulla aina pelatessa uusi kortti pöytään.

Pelistä puuttuu animaatiot, mikä tekee tekoälyä vastaan pelaamisesta melko epäselvää pelaajalle, joka ei tunne kortteja ulkoa.

Kurssilla toteutettu tekoäly on hyvä, mutta sitä voitaisiin suoraan parantaa muuttamalla osaa metodeista tai toteuttamalla uusi tekoäly kokonaan eri periaatteella. Esim. pelatessa kortteja pöytään niitä kokeillaan satunnaisessa järjestyksessä sen sijaan, että pyrittäisiin optimoimaan resurssien käyttöä pelaamalla paras yhdistelmä. Hyökkäysmetodia voitaisiin myös optimoida lisää - lisäämällä keino jolla usean minionin yhdistelmällä voidaan tappaa minioneita, voitaisiin satunnainen hyökkääminen todennäköisesti poistaa kokonaan.