

Viikkoraportti 1

Olen aloittanut projektin tutkimalla erilaisia mahdollisia tapoja toteuttaa Tetristä mahdollisimman hyvin pelaava tekoäly. Lähteitä on kohtuullisen vähän (kuten yksi löytämistäni nettisivuista asian ilmaisi, tetris-tekoälyistä kiinnostuneita on keskimäärin yhtä paljon kuin Kuussa on asukkaita), mutta se toisaalta helpottaa erilaisten ratkaisumallien kartoittamista. Pääasiallisesti olen löytänyt kaksi erilaista lähestymistapaa: heuristinen brute force -haku, jolla etsitään käsiteltävälle palikalle paras mahdollinen paikka, ja geneettinen algoritmi joka analysoi erilaisten pudotuspaikkohjen paremmuutta. Heuristinen haku lienee helpompi toteuttaa, mutta noin yleensä ne eivät näyttäisi toimivan aivan yhtä hyvin kuin elegantimmin rakennetut tekoälyt. Oleellinen ongelma niissä on heuristiikan kehittäminen sellaiseksi, ettei se aja tekoälypelaajaa pussiin esim. estämällä tekoälyä pudottamasta pitkiin, i-palikalle sopiviin koloihin muita palikoita silloin, kun i-palikkaa ei ole saatavilla.

Löysin myös pari erilaista mielenkiintoista tapaa rakentaa sääntöjä joita tekoälyn on noudatettava, esim. kentän reunojen suosiminen palikoita sijoittaessa, mahdollisimman matalan sijainnin suosiminen, jne.

Olen päättänyt toteuttaa tekoälyn luomalla mahdollisista siirroista siirto-olioita, jotka tallennetaan kekkoon, jotta niistä paras saadaan nopeasti käyttöön. Tämän pitäisi olla kohtuullisen hyvä ratkaisu, sillä pahimmassakin tapauksessa eri vaihtoehtoja kunkin muodostelman sijoitukselle on alle 40. Siirron kriteereinä ovat muodostelman lopullisen sijainnin korkeus (korkeudesta tulee miinus pisteitä), muodostelman toisia palikoita ja seiniä koskevien sivujen määrä (jokainen muodostelman muodostavan palikan ulkosivu on oma sivunsa, jolloin esim. tyhjiin peliin pudotettu t-palikka koskettaa kulmassa seiniä neljällä sivulla), sekä siirron avulla muodostuvat rivit (riveistä saa kymmenkertaiset pisteet, sillä niillä on suurin paino). Aion siis toteuttaa näitä kriteerejä käyttävän tekoälyn, mutta varsinainen koodaaminen on vielä alkutekijöissään. Kirjoitan tietenkin myös oman kekototeutukseni kun tekoäly itse toimii. Tällä hetkellä keon toteuttavaan luokaan on implementoitu javan PriorityQueue.

Tällä viikolla olen keskittynyt enemmän algoritmin hahmottelemiseen ja tarpeellisten luokkien sekä ominaisuuksien luomiseen. Itse tekoälylle ei ole vielä yhtään testiä, sillä se ei vielä tee mitään, mutta Siirto- ja Siirtokeko-luokille on omat testinsä.

Lisäksi olen toki pelaillut tetristä ja koettanut selvittää, miten itse päättelen palikoiden parhaan paikan ja millaisia sääntöjä tästä voi johtaa, sekä koettanut emuloida erilaisia tekoälyjä pelaamalla tiettyjä sääntöjä noudattaen. Lisäsin myös tetris-koodini ja kaikki sen testit github-repositorioon kehittelyä varten.

Epäselvää on tällä hetkellä se, miten integroin tekoälyn varsinaiseen peliin siten, että se osaa tehdä siirtonsa oikealla hetkellä, mutta tämä lienee suhteellisen suoraviivaista, kunhan sinne asti pääsen. Toinen hieman epäselvä asia on se, miten toteutan seiniä/palikoita koskettavien sivujen laskemisen, koska tarkoitushan tietenkin on olla mahdollisimman tehokas. Siihen varmaankin pätee jokin verkkoalgoritmin tapainen, mutta en ole ehtinyt paneutua asiaan vielä mitenkään merkittävästi. Erilaisia vinkkejä otetaan mielellään vastaan.