

Testausdokumentti

Yksikkötestauksen kattavuusraportti.

ristinolla					
Files	IE	●	●	●	Coverage
testit	507	499	0	8	98.42%
merkinta.py	25	17	0	8	68.00%
pelaaja.py	17	17	0	0	100.00%
ruudukko.py	29	22	3	4	75.86%
tietokonepelaaja.py	447	375	15	57	83.89%
transponointitaulukko.py	49	36	0	13	73.47%
Folder totals (10 files)	1074	966	18	90	89.94%
Project totals (10 files)	1074	966	18	90	89.94%

Sovelluksesta on testattu kaikki luokat, jota sovelluksessa käytetään lukuun ottamatta seuraavia poikkeuksia. Testistä pois jätettiin main.py koska se sisältää lähinnä käyttöliittymään ja pygamen hallintaan liittyvää koodia. Lisäksi transponointitaulukko.py ja merkinta.py -luokat on jätetty pois testeistä koska ne liittyvät transposition-taulukon toimintaan ja siten ilmeisesti ”ylikurssia”. Tästä sovittu ohjaajan kanssa.

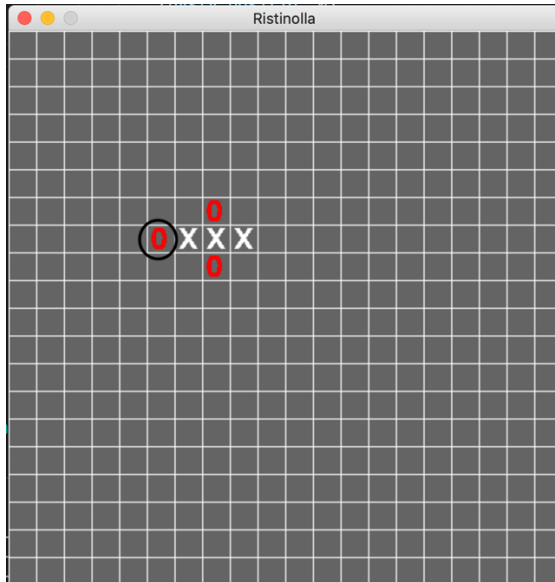
Yksikkötestauksessa on käytetty Pytest:iä ja muuhun koodin laadunvarmistamiseen Pylint:iä (pisteet **8,57/10**).

Kaikki testin tähän mennessä on tehty automaattisesti. Yksikkötestaus ajetaan aina kun githubiin työnnetään muutoksia. Pylint ajetaan paikallisesti.

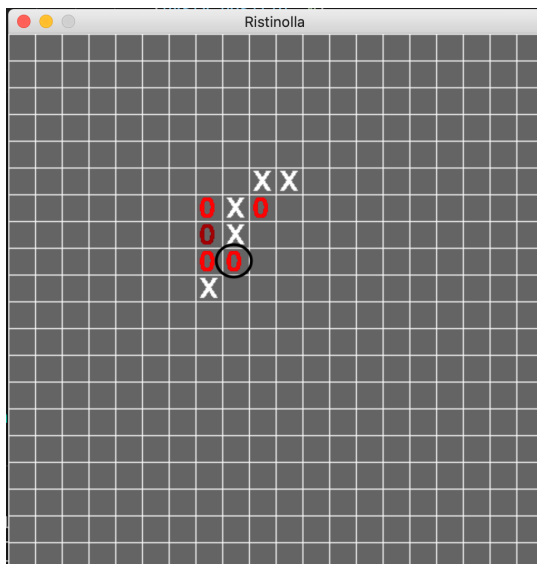
Sovelluksen ”oikeellisuutta”, eli sitä toimiiko algoritmi oikein, testataan Pytesteillä. Koska minimax-algoritmin toteutus on sidottu tietokonepelaajan toteutukseen missä Tietokonepelaajan metodi siirrä() on vastuussa seuraavan siirron palauttamisesta voimme suoraan testata tämän metodin palauttamia arvoja pelin ”oikeellisuuden” testaamiseen. Käytännössä tietokonepelaajan taulukkojen arvot alustetaan yksikkötestissä sopivalla pelitilanteella ja katsotaan antaako metodi ”oikean” tuloksen. Koska algoritmin toiminnassa ei juurikaan käytetä mitään sattumanvaraisia numeroita, poikkeuksena zobrist-taulukon avainluvut, metodin tulisi palauttaa samalle tilanteelle aina sama vastaus.

Algoritmin oikeellisuutta testattiin seuraavilla tilanteilla:

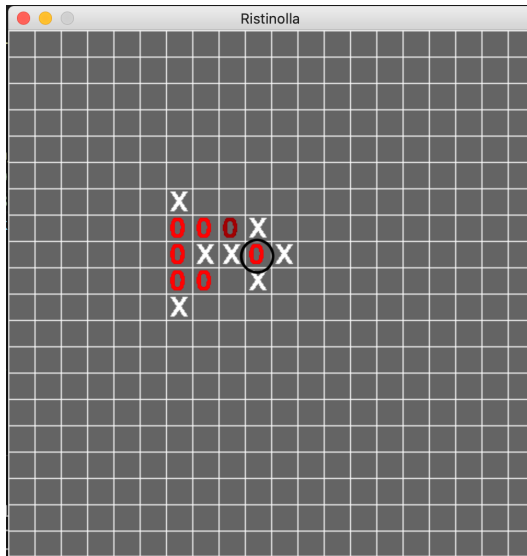
1. Puolustus vapaata kolmen suoraa vastaan. Tietokone palauttaa hyväksytysti (5,7).



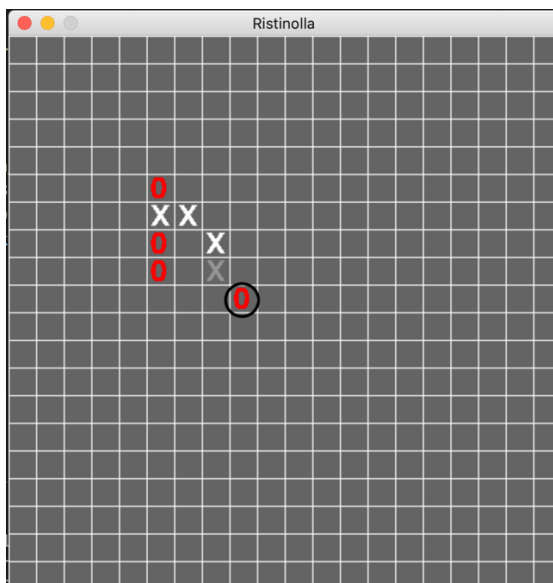
2. Tietokone puolustaa pelaajan yritystä tehdä kaksi kolmen suoraa. Algoritmi palauttaa hyväksytysti (8,8).



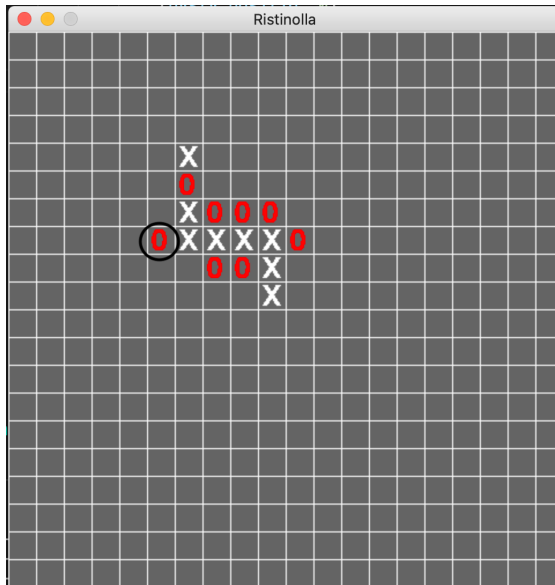
3. Tietokone puolustaa pelaajan yritystä tehdä kaksi kolmen suoraa vastaan. Tietokone palauttaa hyväksytysti (9,8).



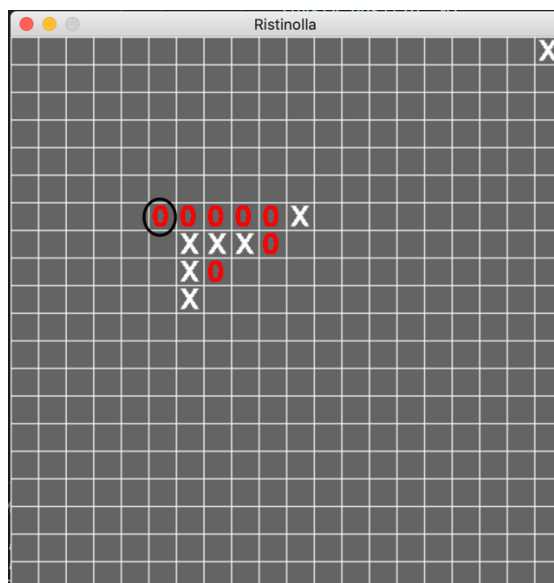
4. Tietokone puolustaa pelaajan yritystä tehdä kaksi kolmen suoraa vastaan. Algoritmi palauttaa virheellisesti (5,9).



5. Tietokone estää vastustajasta tekemästä viiden rivin neljästä. Algoritmi palauttaa hyväksytysti (5,7)



6. Testaa, että tietokone osaa tehdä viiden rivin neljästä. Metodi palauttaa hyväksytysti (5,6).



Peliä testattiin myös sattumanvaraisesti pelaamalla ja siinä ilmeni hieman ongelmia algoritmin kanssa. Käytännössä jos pelissä kummallakin pelaajalla on neljän suora ja tietokonepelaajan tulisi viimeistellä viiden suoransa se poikkeuksetta luulee, että se on hävinnyt ja joko alkaa sattumanvaraisesti laittamaan merkin johonkin tai sitten se alkaa puolustamaan vastustajan suoraa vastaan. Kummassakin tapauksessa se häviää pelin kun sillä on varma voitto. Alla olevassa kuvassa on en esimerkki jossa tietokone luovuttaa ja asettaa merkin sattumanvaraiseen paikkaan. Tälle käyttäytymiselle ei löytynyt ratkaisua monesta yrityksestä huolimatta.

