

Testausdokumentti

Yksikkötestaus

Yksikkötestaus on tehty Unittest-sovelluskehityksen avulla kurssin ohjeita mukaillen. Kaikki ohjelman toiminnallisuuteen vaikuttavat moduulit (pelilogiikka, heuristiikka, minimax-algoritmit, iteratiivinen syventäminen ja tekoälyn siirtovalinta) on testattu automatisoiduilla testeillä, jotka arvioivat onko saatu tulos ennakkoon määritellyn mukainen.

Testien haaraumakattavuus on 96% ja kattaa kaikki oleelliset toiminnot, kattamatta jääneet haarat liittyvät lähes poikkeuksetta varmistuksiin, joilla estetään ohjelman kaatuminen ja joiden ei ikinä kuuluisikaan aktivoitua.

Testit voi toistaa Poetryllä komennoin:

Poetry install

poetry run pytest

poetry run coverage run --branch --omit="tests/*" -m pytest

poetry run coverage html

Coverage report: 96%

Files Functions Classes

coverage.py v7.12.0, created at 2025-12-19 12:38 +0200

File ▲	Statements				Branches			Total
	coverage	statements	missing	excluded	coverage	branches	partial	coverage
AI / ai_move.py	91%	33	3	0	80%	10	2	88%
AI / evaluate.py	100%	25	0	0	100%	18	0	100%
AI / iterative_deepening_midgame.py	100%	25	0	0	83%	12	2	95%
AI / minimax_endscore.py	100%	29	0	0	100%	12	0	100%
AI / minimax_midgame.py	100%	45	0	0	100%	22	0	100%
AI / reorder.py	100%	22	0	0	100%	8	0	100%
reversi / board.py	96%	79	3	0	89%	38	2	94%
Total	98%	258	6	0	93%	120	6	96%

coverage.py v7.12.0, created at 2025-12-19 12:38 +0200

Yksikkötesteillä testattiin muun muassa seuraavat kokonaisuudet:

pelilogiikka: sääntöjen noudattaminen, siirtojen laillisuus, vuoron vaihto ja pelin päättymisen tunnistaminen

heuristiikka: testattiin että palauttaako heuristiikka odotetun arvon kaikilta osa-alueiltaan.

minimax_midgame: oikeaoppinen toiminta, mm. alpha-beta-karsinta, voiton tunnistus, keskipelissä ja hash-transpositiotaulun toiminta.

minimax_endscore: palautuuko paras mahdollinen lopputulos ja lasketaanko pisteet oikein.

iteratiivinen syventäminen: toiminta ja aikarajat.

ai_move: siirtovalinnan toimivuus.

Testisyötteinä käytettiin muun muassa lukuisia manuaalisesti tai reversi-analysaattorilla luotuja pelitilanteita joista tiedetään varma lopputulos ja joiden avulla pystyttiin esimerkiksi määrittämään, että minimax_midgame palauttaa korkean arvon, kun peli on voitettavissa ja myös että alpha-beta-karsinta toimii. Myöskin transpositiotaulun toiminta on tarkistettu ennalta lasketun pelitilanteen avulla.

Benchmark-testaus

Ohjelmaa testattiin peluuttamalla satoja kertoja useita eri painotuksilla toimivia tekoälyjä vastakkain ja tulosten perusteella nykyinen heuristiikkapainotus on todennäköisesti vahvin mitä yksinkertaisilla muutoksilla tähän ohjelmaan pystytään luomaan. Otteluparia ei myöskään voitettu kertaakaan kevyemmällä heuristiikalla tai satunnaisilla siirroilla, kun pelattiin kaksi erää värejä vaihtaen. Lisäksi eri aikarajojen käyttö iteratiiviselle syventämiselle osoitti, että syvemmälle laskeva heuristiikka voittaa tai vähintään pelaa tasan jokaisessa testitapauksessa. Tämä havainto myös puoltaa sitä, että heuristiikka toimii ainakin jossain määrin.

Manuaalinen testaus

Manuaalisella testauksella varmennettiin pelin toimivuus laillisten siirtojen listauksen, virheellisten syötteiden käsittelyn sekä pistelaskennan oikeellisuuden osalta. Lisäksi havainnoitiin, että peli tunnistaa oikein tilanteet, joissa peli loppuu tai vuoro siirtyy. Tekoälyn toiminnan osalta havainnoitiin, että siirrot ovat heuristiikan mukaisia ja odotettuja, että pakotetut voitot löydetään, peliarvio palautetaan oikein ja että AI osaa vaihtaa oikeaan moodiin eri pelivaiheissa riippuen joko jäljellä olevista siirroista tai pelitilanteesta.