

Aineopintojen harjoitustyö: Algoritmit ja tekoäly – Toteutusdokumentti

Miinaharava

Rakenne:

- `__main__` - Tästä alkaa suoritus. Tarkoituksena on käynnistellä sovellusta käyttäjän toiveiden mukaan.
- `cmdline` – Muuntaa komentorivin argumenteiksi, joita voidaan käyttää ohjelmassa
- `app` – Hoitaa yksittäisen pelin suorituksen annettujen argumenttien mukaan. Huolehtii pelilaudan, tekoälyn ja käyttöliittymän alustuksen.
- `board` – Pitää huolen pelilaudasta. Kertoo vain avatut alueet ja käyttäjän asettamat liput.
- `tui` – Tekstikäyttöliittymä hoitaa tulosteet ruudulle ja syötteiden luvun näppäimistöltä. Myös tyköälyä käytetään käyttöliittymän kautta.
- `bot` – Tekoäly, jolta voidaan kysellä vinkkiä. Ottaa laudan kuvan ja palauttaa miinojen sekä vapaiden paikkoja. Myös tarvittaessa tekee arvauksia.

Toteutetut algoritmit:

- `Simple` – Tutkii onko täsmääkö numerolaatta naapurissa olevien miinojen ja vapaiden laattojen kanssa niin että loput peitetyt voidaan merkitä

- DSSP – Tutkii voidaanko kahden vierekkäisen numerolaatan leikkauksessa päätellä olevan miinoja ja ne huomioiden onko avaamattomat laatat siten varmuudella vapaita.

Osa lähdekoodista joka toteuttaa DSSP algoritmin:

```
for tile1, tile2 in self.get_pairs():
    unknowns1, minecount1 = self.get_unknowns_and_minecount(tile1)
    unknowns2, minecount2 = self.get_unknowns_and_minecount(tile2)
    # Kun 1. alueen miinoista vähennetään vain 1. alueella olevat
    # tuntemattomat laatat saadaan vähimmäismäärä miinoille yhteisellä
    # alueella.
    common_minecount = minecount1 - len(unknowns1-unknowns2)
    # Turha jatkaa jos yhteiselle alueelle ei tarvitse asettaa miinoja.
    if common_minecount < 1:
        continue
    # Vähennetään yhteiset tuntemattomat 2. alueen tuntemattomista.
    unknowns2 = unknowns2 - unknowns1
    # Jos 2. alueelle ei jää tuntemattomia ei tarvitse jatkaa
    if not unknowns2:
        continue
    # Vähennetään 2. alueen miinoista ne jotka on pakko sijoittaa
    # yhteiselle alueella
    minecount2 -= common_minecount
    # Jos 2. alueelle ei jää yhtään miinaa tiedetään kaikki
    # tuntemattomat siellä vapaiksi
    if minecount2 == 0:
        self.safe_tiles |= unknowns2
```

Mitä voisi seuraavaksi toteuttaa?:

- Ratkaisija, joka kävisi läpi kahden vierekkäisen sijaan pidempiäkin ketjuja.

Vertailut:

Ratkaisija-algoritmejä voidaan vertailla satunnaisilla tai vartavasten tehdyillä pelilautoilla. Ohjelmalla on helppo ajaa isokin määrä erilaisia pelilautoja ja selvittää miten hyvin ja miten nopeasti algoritmit toimivat.

Laajojen kielimallien käyttö:

Laajoja kielimalleja ei ole käytetty mihinkään tässä projektissa.

XXXXXXXXXX
TKT-tutkinto-opiskelija

Toteutusdokumetti (sivu 3/3)
Aineopintojen harjoitustyö:
Algoritmit ja tekoäly
18.02.24

Lähteet:

<https://dash.harvard.edu/handle/1/14398552>