

Toteutusdokumentti

Ohjelma on jaettu muutamaasi eri osaan: BotRunner, Bot, DecisionMaker ja Astar luokat ovat ohjelman ydin. Main metodista valitaan millainen botti otetaan käyttöön, tosin vain yksi botti on tällä hetkellä olemassa: basic. Tämän jälkeen luodaan BotRunner, joka pyörittää bottia, ja tekee kaiken tarpeellisen, jotta botille annetaan tarvittavat tiedot, ja että serveri kommunikaatio toimii. Bot tekee liikkumispäätöksen perustuen DecisionMaker luokan tekemään prioriteetti päätökseen, ja Astar luokan palauttamaan reittivalintaan perustuen. Tämä päätös annetaan BotRunnerille, joka lähettää viestin serverille, ja odottaa sieltä vastausta joka sisältää seuraavan vuoron tiedot. Tätä jatkuu kunnes peli on ohi.

Aikavaativuudet kaikilla tietorakenteilla vastaavat Tietorakenteet ja algoritmit kurssilla esitettyjä. Hajautukseen pohjautuvilla luokilla (MyHashMap ja MyHashSet) on $O(n)$ sekä lisäys- että hakunopeus, sillä jos kaikilla alkioilla on sama hajautus niin ne menevät linkitettyyn listaan. Keksimäärin nopeus on $O(1)$. Tilavaativuus on $O(n)$.

Linkitetyllä listan haun aikavaativuus $O(n)$, mutta lisäämisen aikavaativuus $O(1)$. Näin käy, koska listalla on tieto siitä, mikä on listan viimeinen olio, joten siihen saa helposti lisättyä uuden perään. Tilavaativuus on $O(n)$.

Minimikeon aikavaativuudet ovat $O(\log n)$ lisäyksellä, ja poistamisella $O(n)$, heapifyn takia. Tilavaativuus on $O(n)$. MyPriorityQueue kasvattaa dynaamisesti kokoaan, joten 21, 41, 81, 161... lisäys vie $O(n)$ aikaa, kun täytyy kasvattaa listan kokoa, ja muuten $O(n)$.

Parannuksia voisi tehdä botin peli logiikkaan. Tällä hetkellä se tekee niitä muutaman eri muuttujan perusteella, kuten etäisyys, omat elämänpisteet ja vastustajien elämänpisteet, mutta jos olisi ollut enemmän aikaa suunnitteilla oli käyttää kehittyneempää päätöksenteko algoritmia botissa. Tällä hetkellä se ei pärjää muita pelaajia vastaan, sillä muut botit käyttävät paljon tehokkaampia algoritmeja.

A* algoritmi suoritetaan tällä hetkellä kerran joka vuorolla. Tätä voisi parantaa esim. Luomalla aluksi jonkinlaisen etäisyys matriisin kartasta, ja tämän jälkeen hakeminen olisi nopeampaa. Tämä ei tosin huomaisi muita pelaajia, ja jonkinlainen muu ratkaisu täytyisi tehdä. Yksi idea mitä mietin oli, että jokainen kaivos ja vastustaja loisivat eräänlaisen "lämpö" alueen ympärilleen, jonka avulla botti kulkisi kohti suurimpia lämmönlähteitä. Tätä ei kuitenkaan ollut tarpeeksi helppo/nopea kehittää joten se jäi vain ajatuksen asteelle.

Lähteet

<https://cs.stackexchange.com/questions/56176/a-graph-search-time-complexity>

https://en.wikipedia.org/wiki/A*_search_algorithm

Tietorakenteet ja Algoritmit-kurssin materiaali, vuoden 2017 kevät.