

Ussimängu iseõppiv AI

Töö eesmärk ja tehisintellekti kasutamise vajadus

Eesmärk oli luua tehisintellekt, mis ise õpib ussimängu mängima. Tahtsime kasutada tehisintellekti, et reaalselt näha, kuidas see õppimine toimib. Samuti oli ka isiklik huvi selle vastu.

Kasutatud ideed (nii kursuse materjalidest kui mujalt) koos viidetega

Alus ussimängule saadud:

<https://github.com/rajatdiptabiswas/snake-pygame>

Info geneetilise algoritmi kohta:

<https://theailearner.com/2018/11/09/snake-game-with-genetic-algorithm/>

Iga autori enda panuse kirjeldus

Sander - Esialgse geneetilise algoritmi koostamine (närvivõrk, individid, generatsioonid jne.) ning selle silumine, aluskoodi funktsioneerimine

Albert - Parameetrite peenhäälestus populatsioonides, närvivõrkudes ja skoorimises, geneetilise algoritmi silumine

Programmi testimisvõimalused ja vähemalt mõned testimistulemused

Programmi on võimalik testida läbi main.py faili.

Selleks tuleb laadida kõik seotud failid ühte kausta ning jooksutada main.py. See fail tegeleb ise teiste failide importimisega.

Tulemuseks on aken, kus on näha, kuidas AI õpib ussimängu mängima. Terminalis on näha iga generatsiooni igat kromosoomi väärtust, mis ta mängides sai.

Töö käigu kirjeldus

Esialgu hankisime endale ussimängu aluse, kuhu hakata looma geneetilist algoritmi kasutavat AI-d, millest me tegime funktsioonid mängu mängimiseks. Seejärel uurisime, mida kujutab endast geneetiline algoritm. Alustasime närvivõrgu loomisega, järgmisena otsisime ja mõtlesime millist funktsiooni kasutada ühe indiviidi tulemuse hindamiseks. Jõudsime järeldusele, et me anname positiivselt punkte "toidu" söömise eest ja negatiivseid punkte kui ta läheb seinale, iseendasse või kui ta läheb seinale lähedale. Siis lõi "klassi", kus indiviidi informatsioon ja meetodid hakkasid paiknema. Programmi käivitades luuakse 25 indiviidi populatsioonis, seejärel lastakse neil kõigil mängida, kus nad saavad endale skoori. Peale kõigi mängimist valitakse nende seast parimate tulemustega individid ja neid pannakse kokku laste tegemiseks nii et iga kaalu jaoks, mis on lapsel vaja, saab ta suvaliselt ühe oma vanema kaalu. Seejärel toimub mutatsioon lapse ühes kaalus ja nii tehakse 50 uut last ja on järgmine populatsioon olemas, millel sama algoritmi kasutada.

Üheks probleemiks oli ussimängu alus, mille me võtsime, sest seal mängis kasutaja ussimängu, mitte AI. Teisisõnu, pidime vahetama mängijat. Peamine osa mängu tööst oli ühes suures *while* tsüklis, mille pidime eraldama/lõikama erinevatesse funktsioonidesse, et AI saaks seda alust üldse kasutada ning ka meil oleks algoritmi kirjutamisel selge alus, millest koostada.

Puudulikuks jäi ka indiviidi skoorimise strateegia, kui oleksime natuke rohkem seda peenhäälestada saanud, oleksime palju paremaid tulemusi näinud kiiremini. Praeguses implementatsioonis on algoritmil raske välja saada lõpmatustest ringidest.

Lisainformatsiooni projekti kohta:

GitHub: https://github.com/albertunn/snake-pygame_AI/