

Tegu on ussimänguga, mis peaks suutma mängida seda võimalikult hästi. Oma töös olen kasutanud pygame moodulit, et teha graafiline ussimäng. Kui on soovi seda kasutajana proovida, siis piisab kui koodis äravahetada

„skoor = mäng("arvuti)"“   ->   „skoor = mäng("inimene)"“

vastu ja siis mängu alates saad kontrollida seda ka nooltega. Ei teinud sellele eraldi nuppu kuna puudus vajadus ise seda mängida, sest projekt keskendub arvuti poolt mängitud mängule.

Esimese asjana andsin Alle ette käigud, mida ta teha võib ehk vasak, parem, üles ja alla. Proovides paar korda sain kohe aru, et on vähemalt 25% iga käik, et ta tapab end ära, sest ma ei keela tal tagasiliikuda ja siis sööb ennast. Siiski proovisin mõned korrad veel seda ja juhtus ime, ilma, et oleksin midagi lisanud, siis suutis uss koguda 4 punkti, mille tõenäosus peaks olema hämmastavalt väike (keskmiselt on maiustus u 60 käigu kaugusel, iga käik on keskmine suremisvõimalus u 30%, ja seda 4 korda ehk  $1/6.66 \cdot 10^{38}$  peaks olema võimalus see läbiteha, kui mu matemaatika klapib). Seejärel andsin ussile ette reeglistiku, kuidas end mitte tappa ehk kirjutasin, et ta ei tohi seina süüa ega iseennast (meetodid, kasOnSein ja kasOnUss kontrollivad seda mängus). See tegi ussi selliseks, kes lihtsalt tiirutas mööda kaarti ringi, võis tunde vahel isegi nii teha, väga tark polnud, lõpuks kui oli piisavalt suur, siis mingi hetk keeras end lõksu ja kuna ühtegi käiku teha polnud, siis sõi end ja kaotas. Pikim uss, mida ma selle meetodiga sain, oli vist 40 punkti. Seejärel hakkasin vaatama, mis käigud võiksid olla parimad (seda kontrollib meetod parimadKäigud, ideena kasutasin 7. praktikumis tehtud trips-traps-trulli, kus kasutati ka hinnanguid mängija loomisel). Uss muutus palju paremaks, liikus maiustuse juurest otse maiustuse juurde, kuid varem või hiljem sõi end ikka lõksu. Selle vältimiseks üritasin kasutada rekursiooniga käikude ettevaatamist, kuid see tegi ussimängust powerpoint esitluse kui sedagi. Mida rohkem käike ma ettevaatasin, seda aeglasemaks see muutus ja kuna vaid ühest või kahest käigust polnud kasu, siis manuaalselt panin kontrollima ühte käiku (see eemaldab trepina liikumisel tekkinud aukudesse minemise, keskmist eluiga suurendas u 30 punkti võrra) ja mõtlesin, kuidas seda veel parandada. Selleks tegin siis lõpuks 2 tsüklit, mida uss läbima hakkab. Kõigepealt lasen tal koguda 48 punkti kiireima teega (48

punkti on ta saanud u 90% kordadest seega tundus hea arv, kui ei saa seda kätte siis soovitan uuesti proovida) ja seejärel hakkab ta tsükleid läbima, kus ta teeb ringi peale kogu kaardile (funktsioonid tsükkel ja tsükkel2). Ideaalis peaks ta nüüd siis koguma nii palju punkte, et lõpuks ei mahu kaardile ära ja sööb iseennast. Probleem tekkis hetkel selles, et olin kaardi nii suure võtnud ja selle saavutamiseks läheb väga kaua aega. Pikim punktisumma, mis ma oodata viitsisin sellega, oli 205 punkti vist, kui õigesti mäletan. 2 erinevat tsüklit pidin kasutama, kuna kui uss läbib ühe korra terve kaardi, siis ta suund ülesse tagasijõudes on teistpidine ja siis lasen tal kasutada teist tsüklit ja need siis omavahel vahetuvad kordamööda. Ainus asi, mis hetkel ma tahaks parandada tulevikus on see, et optimiseerida neid tsükleid näiteks, kui alumisel real ei ole ühtegi ussi osa ega maiustust ennast, siis ta lihtsalt liigub alla, kuni tuleb üks nendest (eeldatavasti maiustus siis) ja jätkab tsükliga, see kiirendaks lahenduskäiku väga palju (see tuli pähe alles viimasel hetkel ja kuna ei saanud esimese hooga seda õigesti tööle, siis ei esita seda).

Panus – Esialgselt oli plaanis graafiline pool võtta internetist ja tegeleda ise vaid AI-ga, kuid kuna ei leidnud sobivat, siis on põhimõtteliselt kõik minu enda kirjutatud (erandina paar koodirida, mis tuli stackoverflowst, näiteks maksimumi leidmine dictionaryst).