UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Daniel Rižnar, Uroš Kosič

Umetna inteligenca 2

SEMINARSKA NALOGA
POROČILO

Mentor: Martin Možina

Ljubljana, 2011

Kazalo

Povzetek

Namen dokumenta je, predstaviti naše delo, ki se je izvajalo v sklopu vaj pri predmetu Porazdeljene inteligentne programske tehnologije. Cilj vaj je bil implementirati znanja in ideje, ki smo jih pridobili pri predmetu. Implementacija je v obliki agenta, ki sodeluje z drugimi agenti pri igranju igre Capture The Flag (CTF). Asistent pri predmetu nam je pripravil simulirano okolje agenta, v katerem se igra odvija.
V tem dokumentu se bomo dotaknili problemov, kot so raziskovanje prostora z več agenti, sodelovanje med agenti, iskanje zastave, ter razne ofenzivne in defenzivne taktike pri sami igri.
Pri raziskovanju prostora je cilj, podobno kot pri raziskovanju z enim samim agentom, minimizirati skupen porabljen čas. V našem primeru je cil, da agenti v čim krajšem času odkrijejo zastavo. Glavni problem, ki ga je potrebno rešiti v kontekstu raziskovanja z večimi agenti je izbira primernih smernic za posameznega agenta, tako da istočasno raziskujejo različne predele prostora. Tega smo se lotili tako, da vsak agent oceni ceno poti do ciljne točke in njen prispevek. Ob vsakem morebitnem srečanju pa si izmenjajo informacijo o prostoru, ki so jo do tistega trenutka pridobili.

Za oblikovanje tega dokumenta je bil uporabljen sistem LaTeX.

Ključne besede:

Agenti, sodelovanje, CTF, raziskovanje, algoritmi, AI.

Poglavje 1

Opis naloge

RPG (role-playing game) ali igra igranja vlog je izraz za igre, v katerih igralci prevzamejo vloge namišljenih likov, postavljenih v domišljijsko okolje. Bodisi z neposrednim igranjem, bodisi z opisovanjem njihovih dejanj nato igrajo te vloge v zgodbi, ki je lahko vnaprej začrtana ali pa nastaja sproti. Uspeh ali neuspeh njihovih dejanj določa dogovorjen sistem pravil konkretne igre. Velik problem je ravnotežje različnih likov, saj se v večini primerov izkaže, da imajo nekateri liki prednost pred ostalimi in lažje zmagujejo v neposrednih dvobojih.

Posamezna RPG igra ponuja izbiro med več različnimi bojevniki, vsak ima svoje prednosti in slabosti. V tem poglavju bomo opisali konkretne značilnosti naše igre in predstavili problem.

1.1 Značilnosti igre

V igri sodelujeta dva igralca, vsak s svojim bojevnikom. Bojno polje je plošča poljubne velikosti, razdeljena na kvadratno mrežo. Bojevnika izmenično izvajate v naprej definirane akcije. Cilj igre je pokončati nasprotnika, preden on pokonča tebe. Vsak bojevnik ima na izbiro v naprej določene akcije, ki jih opisuje tabela 1.1. Posamezni bojevnik je opisan z atributi, ki jih podaja 1.2.

1.2 Bojevniki

Za potrebe seminarske naloge, smo se omejili le na dva bojevnika - Tank in Vojak. Značilnosti obeh smo določili kot v tabeli 1.3.

Tabela 1.1: Akcije bojevnikov

Akcija	Opis Akcije
Move	Akcija pomeni premik v eno izmed štirih smeri (gor, dol, levo, desno). Vsak premik zahteva določeno količino energije.
Pass	Akcija pomeni počitek, bojevniku se ob počivanju poveča energija.
Fire	Strel s primarnim orožjem. Vsak strel pomeni zmanjšanje določene količine energije in zmanjšanje življenja nasprotnika. Škoda je delno odvisna tudi od naključja.

Tabela 1.2: Atributi bojevnikov

Atribut	Opis Atributa
Life	Celo število, ki podaja preostalo življenje.
Speed	Hitrost, ki opisuje za koliko kock se bojevnik lahko premakne ob premikih.
Energy	Preostala energija bojevnika, ki jo lahko porabi za izvajanje akcij.

1.3 Opis problema

Znan problem v RPG igrah je ravnotežje različnih likov, ki tekmujejo med seboj. Kljub različnim lastnostim, morajo imeti bojevniki enake možnosti za zmago v igri. Naša naloga je bila poiskati konkretne vrednosti atributov za zgoraj opisana bojevnika tako, da bosta igrala čim bolj izenačeno igro. Problem smo ločili na dva dela - simulacijo igranja in optimizacijo izenačenosti bojevnikov. Za simulacijo igranja smo uporabili *minimax* algoritem. Za optimizacijo smo uporabili *genetske algoritme* in jih primerjali s *hill climbing*-om.

Tabela 1.3: Značilnosti bojevnikov

Tip	Trpežnost	Domet	Hitrost	Obnavljanje	Varčnost	Energija
Tank	boljša	boljši	višja	slabše	boljša	višja
Vojak	slabša	slabši	nižja	boljše	slabša	nižja

Poglavje 2

Metode dela

V tem poglavju bomo opisali uporabljene algoritme in načine izvajanja meritev. Na koncu poglavja bomo podali nekaj implementacijskih podrobnosti.

2.1 Simulacija igranja

Za simulacijo igranja igre smo uporabili algoritem *minimax*. Z njim smo poskušali doseči, da igralca igrata čim bolj optimalno igro, tj. v danem trenutku izbereta najboljšo možno potezo.

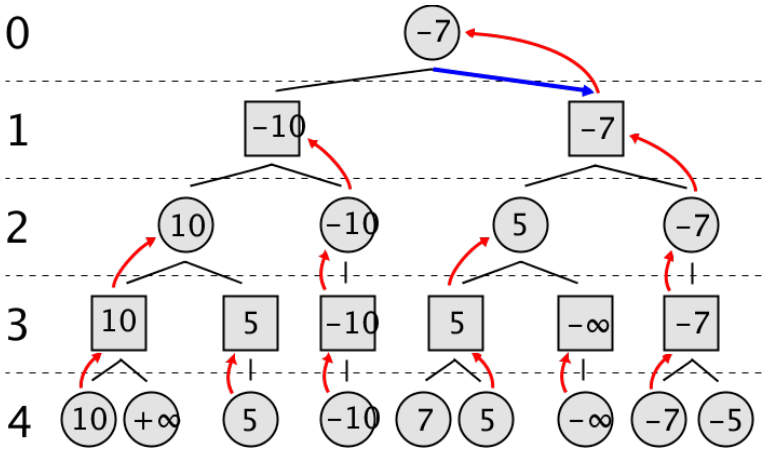
2.1.1 Algoritem minimax

Algoritem se uporablja za minimizacijo možne izgube, pri maksimizaciji možnega dobička. V naši igri igralca izmenjujeta poteze in poskušata izbrati najboljšo svojo potezo oz. maksimizirati svojo vrednost, posledično pa minimizirati nasprotnikovo. Tako glede na trenutno pozicijo identificiramo igralca MIN in MAX. Tisti ki je na potezi predstavlja igralca MAX, nasprotnik pa igralca MIN. Algoritem gradi drevo, kjer je v sodih nivojih (z začetkom v korenu) na potezi igralec MAX, na lihih pa igralec MIN. Na MAX nivojih, igralec vedno izbere stanje, ki maksimizira njegovo vrednost, na MIN nivojih pa igralec izbere stanje, ki minimizira njegovo vrednost (ker predpostavlja, da bo nasprotnik izvedel za njega najslabšo možno potezo). Naslednja poteza je tista poteza, ki jo igralec MAX izbere v korenskem vozlišču. Algoritem je predstavljen na Sliki 2.2. Propagiranje vrednosti vozlišč poteka od spodaj navzgor, zato je potrebno te vrednosti določiti v listih. Ker je prostor preiskovanja prevelik (če bi lahko preiskali celoten prostor, bi v listih vedno imeli samo dve možnosti - zmago ali

poraz), je potrebno višino drevesa omejiti, vrednost v listih pa določiti z neko hevristiko. Naša hevristika je predstavljena na 2.1.

$$val(n) = \begin{cases} -1000 & \text{če je igralec MAX mrtev} \\ +1000 & \text{če je igralec MIN mrtev} \\ +1.0 \cdot MaxLifeRatio - 1.2 \cdot MinLifeRatio + & \\ +0.1 \cdot MaxEnergyRatio - 0.1 \cdot MinEnergyRatio & \text{sicer} \end{cases}$$

Slika 2.1: Hevristika. Členi v tretjem delu funkcije predstavljajo razmerje med trenutnimi in izačetnimi vrednostmi atributov. Intuitivno, hevristika bo ocenila stanje kot dobro, če je življenje MAX igralca visoko, MIN igralca nizko, energija MAX igralca visoka in MIN igralca nizka.



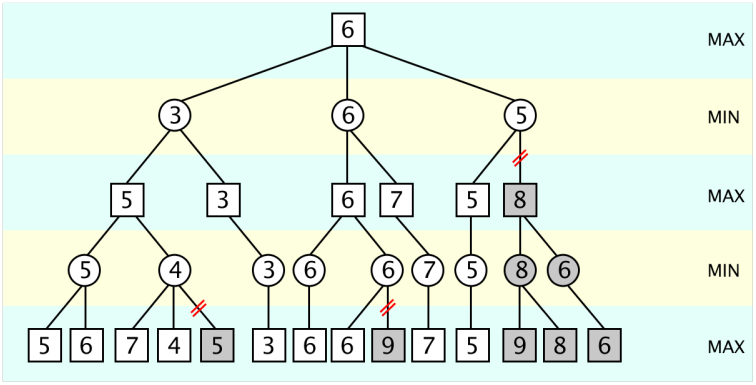
Slika 2.2: Algoritem minimax: slika predstavlja delovanje algoritma minimax. Številke v vozliščih predstavljajo vrednost posameznega stanja z vidika MAX igralca. Puščice predstavljajo propagacijo vrednosti od spodaj navzgor (igralec MIN na lihih nivojih izbira minimalno vrednost, izgralec MAX na sodih nivojih izbira maksimalno vrednost).

2.1.2 Problemi in prilagoditve

V naši igri pomembno vlogo igra tudi naključje. Akcija Fire ima definirano maksimalno in minimalno škodo, ki jo zadane nasprotniku. Vsakič ko bojevnik izbere to akcijo, se vrže kocka, ki z verjetnostjo 0.5 določi stopnjo škode (ali

maksimalna ali minimalna). Zaradi naključnosti, rezultati minimax algoritma ob istih vhodnih podatkih niso vedno enaki. Da bi omilili vpliv naključnosti, je potrebno igro simulirati večkrat, kar privede do povečanja časovne kompleksnosti (sicer za konstanto, a ker se ena simulacija izvaja precej časa, že ta konstanta pomeni veliko razliko). Zato smo implementirali *alfabeta* rezanje, ki ob zagotavljanju enake rešitve kot navaden *minimax* v večini primerov zmanjša število evaluacij listov. Na Sliki 2.3 je prikazan minimax algoritem z alfabeta rezanjem.

Z do zdaj opisanimi pravili igre se je pojavljal problem bežanja. Velikokrat se je zgodilo, da je bojevnik, ki je ugotovil da ne more zmagati, začel bežati pred drugim. Tako je igra trajala zelo dolgo ali pa se sploh ni končala. Zato smo v igro uvedli pravilo, da se bežočega bojevnika kaznuje. Pravilo smo definirali kot: če se nek bojevnik odmakne od drugega in ga v naslednji potezi nasprotnik zadane, se mu hitrost razpolovi. V primeru, da se bežoči umakne izven dosega nasprotnika, se nasprotniku moč zadetka zmanjša za 2^d, kjer je d razdalja med bežočim in zunanjo mejo dosega akcije. Tako bežočega upočasnimo in damo nasprotniku možnost, da ga dohiti oziroma pokonča. Zdaj se bojevniki manjkrat odločajo za bežanje.



Slika 2.3: Algoritem minimax z alfabeta rezanjem: slika predstavlja delovanje algoritma minimax z alfabeta rezanjem. Za najblolj desen rez velja: V vozlišču je na potezi igralec MAX in bo vedno izbral vozlišče z maksimalno vrednostjo (trenutno je to vrednost 6). Na nivoju 1 je na potezi igralec MIN in bo vedno izbral minimalno vrednost (v najbolj desnem vozlišču na nivoju 1 je trenutno to 5). Ker pa že vemo, da ima eno izmed vozlišč na nivoju 1 vrednost 6 in ker bo igralec MIN v najbolj desnem vozlišču zagotovo izbral vozlišče z vrednostjo <= 5 (ker imamo 5 že izračunano), lahko na označenem mestu opravimo rez.

2.1.3 Parametri algoritma

Simulaciji igranja iger je potrebno nastavit nekaj parametrov, ki vplivajo predvsem na hitrost izvajanja in natančnost rezultatov. Eden izmed parametrov je globina minimax drevesa. Globina nam uravnava stopnjo optimalnosti neke poteze, saj z večjo globino drevo razišče večji del prostora. Z večanjem globine, se časovna zahtevnost algoritma eksponentno povečuje. Mi smo se odločili, da bomo uporabljali globino 4

2.2 Optimizacija uravnoteženosti

Za uravnoteževanje bojevnikov smo uporabili genetske algoritme in hill climbing.

2.2.1 Genetski algoritmi

Genetski algoritmi so prilagodljive metode, ki jih uporabljamo za reševanje iskalnih in optimizacijskih problemov. Temeljijo na genetskem procesu bioloških organizmov, saj se zgledujejo po evoluciji v naravi, kjer se populacija neke vrste skozi generacije razvija po načelu naravnega izbora in preživetju uspešnejšega. Genetski algoritmi posnemajo tiste procese v naravi, ki so bistveni za evolucijo. V naravi posamezniki neke populacije med seboj tekmujejo za življenjsko pomembne vire. Hitrejši in pametnejši predstavniki vrste bodo imeli več in boljče vire za preživetje, počasnejši in manj pametni jih bodo dobili težje ali pa sploh ne. Ravno tako se pojavi tekmovanje pri iskanju partnerja za razmnoževanje. Tisti, ki so uspešnejši v preživetju in pri parjenju, bodo po vsej verjetnosti imeli relativno večje število potomcev. Slabši posamezniki jih bodo imeli manj ali pa sploh ne. To pomeni, da se bodo geni dobro prilagojenih oz. ustreznih posameznikov bolj razširili na prihodnje generacije, slabši pa bodo celo izumrli. Kombinacija genov dveh ustreznih staršev lahko privede do pojava super-ustreznega potomca, ki ima boljše lastnosti od obeh staršev. Na ta način se vrste razvijajo in prilagajajo na svoje okolje. Ti algoritmi delujejo po analogiji k temu naravnemu procesu. Delujejo nad neko populacijo osebkov (kromosomov), od katerih vsak predstavlja možno rešitev danega problema. Vsakemu kromosomu se priredi ocena uspešnosti, ki je prilagojena iskanemu problemu. Ustreznejši kromosomi - tisti z boljšo oceno - imajo več možnosti za reprodukcijo od ostalih. Nad trenutno populacijo izvedemo simuliran proces evolucije. Iz naše populacije izberemo podmnožico staršev, ki se razmnožujejo. Tako za našo populacijo dobimo nove potomce, ki

prevzamejo nekaj lastnosti od vsakega starša. Manj ustrezni predstavniki se bodo razmnoževali z manjšo verjetnostjo in tako izumrli. Na ta način se dobre lastnosti razširijo v naslednje generacije. Če smo genetski algoritem dobro zastavili, bo populacija konvergirala k optimalni rešitvi. V nadaljevanju bomo predstavili konkretne značilnosti genetskih algoritmov za našo domeno.

Osnovni pojmi

Osebek je možna rešitev problema. V naši domeni je osebek sestavljen iz dveh bojevnikov, oziroma množice atributov. Osebke imenujemo tudi *kromosomi*.

Geni so posamezni atributi v osebku.

Kriterijska funkcija je funkcija, s katero ocenjujemo osebke. Pri nas je ta funkcija definirana kot $\frac{1}{LifeDiff}$, kjer je LifeDiff razlika med življenjema dveh bojevnikov v osebku, kot jo vrne algoritem za simulacijo igre.

Populacija je množica osebkov. Obdelujemo jo v korakih, ki jih imenujemo *generacija*. Iz osebkov trenutne populacije (*staršev*) tvorimo naslednike (*potomce*), ki pripadajo populaciji naslednje generacije.

Elita je podmnožica najboljših osebkov v populaciji.

Selekcija

Križanje

Mutacija

2.2.2 Hill climbing

Slike

Tabele