

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO



Rok Kek, Matjaž Verbole

Porazdeljene inteligentne programske tehnologije

SEMINARSKA NALOGA
POROČILO

Mentor: Luka Čehovin

Ljubljana, 2011

Kazalo

Povzetek	2
1 Predstavitev simuliranega okolja	3
1.1 Senzorji	3
1.1.1 Prejemanje sporočil	3
1.1.2 Zaznavanje prostora	4
1.2 Akcije agenta	4
1.2.1 Gibanje	4
1.2.2 Pošiljanje sporočil	5
1.2.3 Manipulacija z zastavo	5
2 Problemi na katere smo naleteli	6
2.1 Raziskovanje	6
2.2 Iskanje poti do zastave	7
2.3 Prenos zastave v domačo bazo	8
2.4 Srečanje z drugimi agenti	8
2.4.1 Srečanje z agenti nasprotne ekipe	9
2.4.2 Žrtvovanje agenta	10
Seznam slik	11

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 cilj, 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 L^AT_EX.

Ključne besede:

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

Poglavje 1

Predstavitev simuliranega okolja

Okolje je implementirano v programskem jeziku Java. Okolje omogoča nalaaganje različnih zemljevidov, postavitev zastave in izbiro načinov igranja. V sklopu vaj smo se osredotočili le na način pri katerem skupina agentov v čim krajšem času najde zastavo in jo prinese v domačo bazo. Na enem zemljevidu je tako več skupin agentov, zmaga tista skupina ki prinese zastavo prva. Na sliki 1.1 je prikazano simulirano okolje brez agentov.

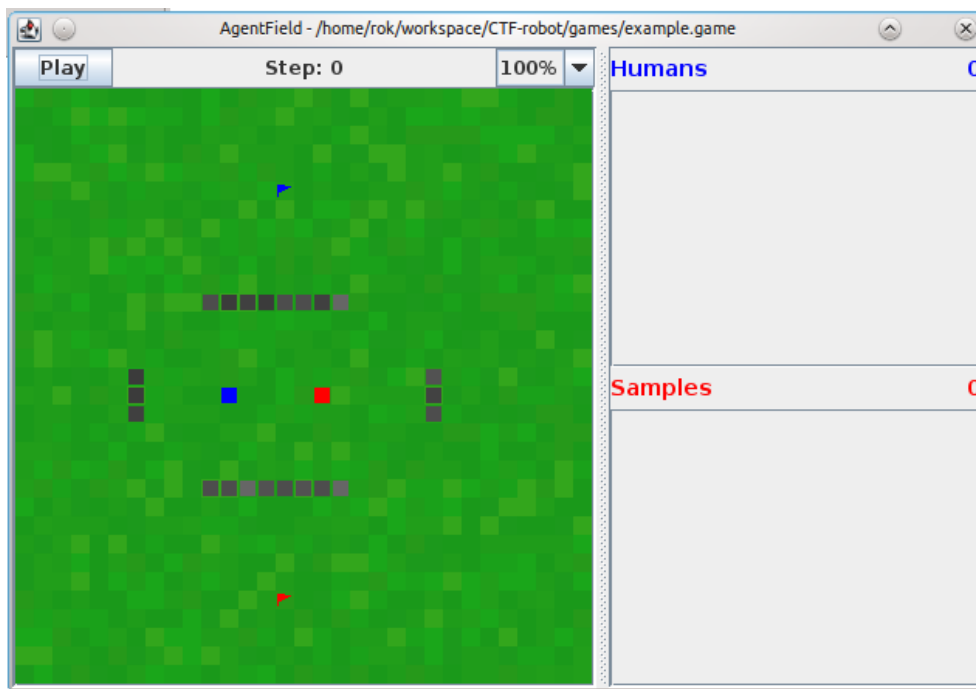
V simuliranem okolju imamo implementirano tudi senzoriko agenta, in akcije ki jih lahko izvaja.

1.1 Senzorji

Senzorika agenta zajema prejemanje sporočil poslanih, od drugih agentov iste skupine in zaznavanja prostora v katerem se giblje. V grafičnem vmesniku simuliranega okolja si lahko vidimo posredovanje sporočil med agenti in pot ki jo je naredil vsak od agentov.

1.1.1 Prejemanje sporočil

Agent sprejema sporočila, ki so mu bila poslana od drugih agentov. Sporočilo je v obliki znakovnega zaporedja ali zaporedja bajtov. Je edini način komunikacije med agenti.



Slika 1.1: Primer simuliranega okolja

1.1.2 Zaznavanje prostora

Zaznavanje prostora je določeno z parametri okolja. Agent zaznava kvadraten del prostora z središčem njegovega položaja. Agent ločuje med različnimi objekti v prostoru, kot so; zid, prazen prostor, nasprotnik, nasprotnikova baza, nasprotnikova zastava, njegova zastava, njegova baza in član njegove skupine z njegovo ID številko.

1.2 Akcije agenta

Agent lahko vpliva na okolje z; svojim gibanjem, z pošiljanjem sporočil in lahko nosi zastavo svoje ekipe.

1.2.1 Gibanje

Agent se lahko vedno premakne v katerokoli od štirih smeri (gor, dol, levo in desno), ne glede na to, kaj je takrat v tisti celici. V primeru da se želi premakniti v celico, ki je označena kot zid, izgubi življenje in se vrne v bazo,

kjer zopet začne z novim preiskovanjem. Podobno se zgodi tudi takrat, če se premakne v celico, kjer je že eden od njegovih ali nasprotnikovih robotov. V tem primeru življenje izgubita oba.

1.2.2 Pošiljanje sporočil

Pošiljanje sporočil je omejeno z razdaljo med agenti. Trenutne nastavitve okolja omogočajo posredovanje sporočil le agentom, ki so znotraj vidnega polja. Sporočilo se pošlje s pomočjo ID številke agenta ki je v isti ekipi. Dolžina sporočila je omejena, prav tako pa so sankcionirani agenti, ki med seboj prekomerno komunicirajo.

1.2.3 Manipulacija z zastavo

Agent lahko pobere zastavo svoje ekipe, in jo odnese v svojo bazo. Če med vračanjem domov agent umre zaradi kakršnega koli vzroka, se zastava v zadnje obiskani celici spusti na tla in ostali agenti potem ko jo najdejo, nadaljujejo od tam naprej. Igra dopušča tudi možnost, da je na enkrat na zemljevidu postavljenih več zastav za eno ekipo.

Poglavje 2

Problemi na katere smo naleteli

2.1 Raziskovanje

Pri raziskovanju prostora je pomembno, da si agenti čimbolj razdelijo prostor, ki ga preiskujejo. Pomembno je tudi, da preiščejo ves prostor, do katerega lahko pridejo (Dokler ne najdejo zastave).

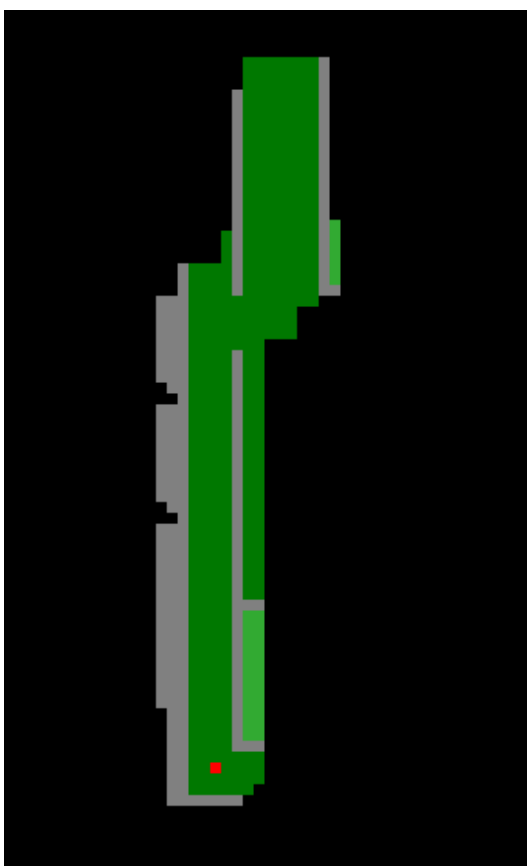
Raziskovanje je implementirano tako, da vse neobiskane celice oceni in nato izbere najboljšo. Ocena se izračuna na podlagi tega, koliko novih celic odkrijemo če obiščemo to pozicijo, ta ocena je skalirana z razdaljo te pozicije od trenutne pozicije. Majhen vpliv na oceno pa predstavlja bljižina zidu. Tem bližje kot smo zidu, tem manjša je ocena. Do najboljše pozicije agent izdelava plan poti s pomočjo algoritma A^* .

Ta strategija se je izkazala kot učinkovita, vendar se včasih izkaže, da na poti izpušča majhne koticke, do katerih mora naknadno dostopati, po seveda veliko daljši poti. Rešitev za ta problem bi bila, če bi agenta prisilili, da raje izbira pot, ki je v ravni liniji.

Da se agenti, ki ne utegnejo komunicirati med sabo (zaradi prevelike razdalje), nebi očali za isto pot, je v samo hveristiko potrebno vnesti nek naključen člen. Ta vpliv pa mora biti seveda manjši od vseh ostalih. Ker je agentov veliko smo implementirali, tako da so nekateri agenti bolj podvrženi naključju in nekateri manj.

2.2 Iskanje poti do zastave

V primeru, ko se zastava agentove skupine pojavi v njegovem vidnem polju ve katera je njegova ciljna celica, ni pa nujno da obstaja pot po raziskanem prostoru, do te celice. V primeru da pot obstaja, je problem trivialen in agent gere po zastavo. V primeru ko pa poti ni, pa je potrebno prostor nadaljne raziskati. Na sliki 2.2 so z svetlejšo barvo označeni predeli, do katerih s trenutnim znanjem agent ne zna priti.



Slika 2.1: Agentova notranja predstavitev sveta

V naši implementaciji agenta se s slednjim problemom soočamo tako, da agent sledi zidu v levo ali desno stran. Tako so nekateri agenti levičarji in drugi desničarji, to se izračuna na podlagi njegove ID številke. Razdelitev agentov na desničarje in levičarje nam reši dve stvari, to da agenti ne sledijo isti poti.

Ter dopušča možnost, da nek agent najde krajšo pot. Seveda vsem agentom, ki jih sreča na poti sporoči svoja spoznanja, tako da se lahko zgodi, da manjkajoči del zemljevida izve od drugih agentov in problem postane zopet trivialen.

2.3 Prenos zastave v domačo bazo

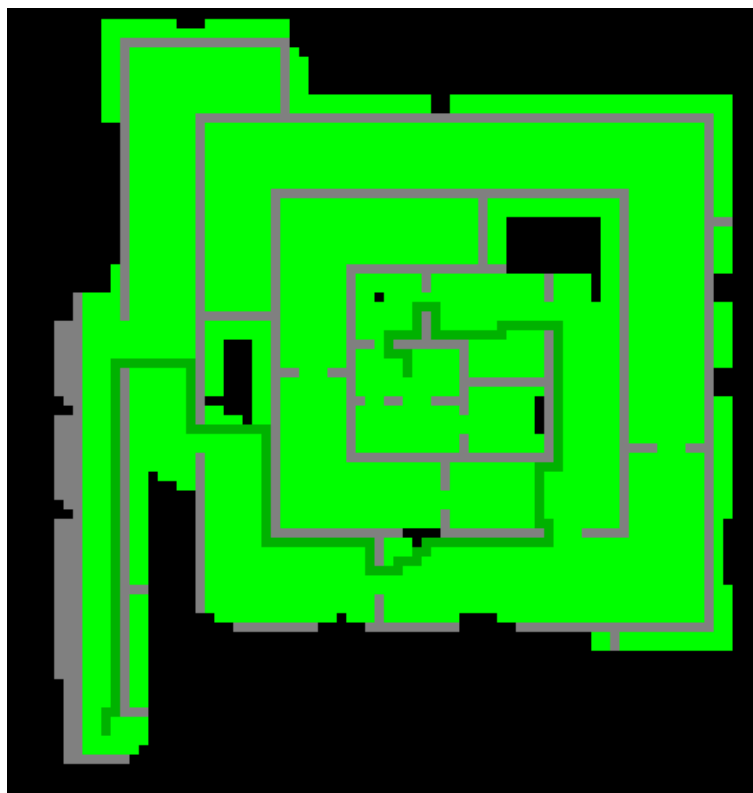
Ko agent pobere zastavo je njegova naloga, da jo prenese v bazo svoje skupine. Pri tem pa lahko naleti na agenta nasprotne ekipe, ki ga po vsej verjetnosti poizkuša ovirati na njegovi poti. Agentu ne preostane drugega, kakor da se mu umika. Vendar pa je gibanje agenta, ki nosi zastavo počasnejše od gibanja agenta, ki zastave nima. Tako da je v tem primeru situacija neizbežna. Pomagamo si lahko le tako, da agenti iste skupine ščitijo agenta z zastavo.

Agent s pomočjo algoritma A^* , naredi plan do baze in ga izvaja. Slika ?? prikazuje izdelan plan Agentu lahko ščitimo na dva načina. Prvi način je ta, da se z agentom, ki ne nosi zastave, poizkušamo zaleteti v agenta nasprotne skupine. Tako ima naš agent z zastavo prosto pot. Drugi način pa je ta, da agent brez zastave le spremlja agenta, ki zastave nima. Ko se nasprotnikov agent zaleti v agenta z zastavo (ob tem seveda oba agenta izgubita življenje), tako lahko preživel agent pobere zastavo in nadaljuje pot proti bazi. Odločili smo se za implementacijo drugega primera ščitenja agentov, saj je lažji in prav tako učinkovit.

2.4 Srečanje z drugimi agenti

Pri srečanju agenta iz iste ekipe, ob seveda prednosti komunikacije in posredovanja medsebojnih informacij, obstaja nevarnost medsebojnega trka. To predstavlja nevarnost, da dva agenta izgubita življenje dva agenta.

Ta problem smo rešili tako, da se agenti z nižjimi ID številkami ne ozirajo na agente z višjimi ID, jim pa sporočijo svojo namero gibanja. Tako se lahko agenti z višjo ID prilagodi agentu z nižjo. Izjema je primer, ko agent nosi zastavo, takrat ima ta agent prednost pred vsemi. Kako drugi agenti reagirajo na konfliktno stanje je odvisno od njihovega notranjega stanja. Če iščejo zastavo



Slika 2.2: Agentov plan od zastave do baze

2.4.1 Srečanje z agenti nasprotne ekipe

Če se agent sreča z agentom nasprotne skupine pomeni, da je v njegovem vidnem polju. V primeru, da je kateri izmed teh dveh agentov “nepazljiv” in se premakne v celico, kjer je drug agent, lahko pride do medsebojnega trka. Če agentova strategija ni usmerjena k uničevanju drugih agentov in s tem posledično tudi samega sebe, je edina logična pot ta, da se agenta poskušata zaobiti in s tem ohranita možnost iskanja vsak svoje zastave.

Naši agenti so sprogramirani tako, da se poskušajo izogibati agentom nasprotne skupine. Ko se nasprotni agent približa našemu agentu na razjo ene celice, se naš agent premakne v nasprotno smer od sovražnega agenta. To pomeni, če naš agent vidi nasprotnika levo od svoje trenutne pozicije, se bo v naslednjem koraku premaknil desno, če tam seveda ni kakšne druge ovire (stena, agent). Izjema je primer ko je agent varovan oziroma ima spremstvo, torej nese zastavo proti bazi. V tem primeru, pa agent ne reagira na druge agente, temveč nese

zastavo naravnost proti bazi.

2.4.2 Žrtvovanje agenta

Agenti ob trku oba izgubita življenje. Agent, ki izgubi življenje izgubi vso informacijo, ki jo je do tistega trenutka pridobil. Tako, da je v interesu celotne skupine, da se agent takim situacijam izogiba. Vendar obstajajo situacije pri katerih, se izplača, da se agent žrtvuje. Primer take situacije je ko agent nasprotne skupine nosi v svojo bazo. Vendar senzorika nam te informacije ne posreduje.

V naši implementaciji agenta smo tako ubali strategijo, da agent, ki je svojo informacijo pred kratkim (določeno s parametrom) predal naprej agentu z nižjim ID, ob morebitnem srečanju z nasprotnikom žrtvuje življenje in upa, da mu je povzročil čim več nevšečnosti.

Slike

1.1	Simulirano okolje	4
2.1	Agentova predstavitev sveta	7
2.2	Izdelan plan	9