

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO



Daniel Rižnar, Uroš Kosič

## **Umetna inteligenca 2**

SEMINARSKA NALOGA  
POROČILO

Mentor: Martin Možina

Ljubljana, 2011

# Kazalo

<b>Povzetek</b>	<b>2</b>
<b>1 Opis naloge</b>	<b>3</b>
1.1 Značilnosti igre . . . . .	3
1.2 Bojevniki . . . . .	3
1.3 Opis problema . . . . .	4
<b>2 Metode dela</b>	<b>6</b>
2.1 Simulacija igranja . . . . .	6
2.1.1 Zaznavanje prostora . . . . .	6
2.2 Akcije agenta . . . . .	7
2.2.1 Gibanje . . . . .	7
2.2.2 Pošiljanje sporočil . . . . .	7
2.2.3 Manipulacija z zastavo . . . . .	7
<b>3 Problemi na katere smo naleteli</b>	<b>8</b>
3.1 Raziskovanje . . . . .	8
3.2 Iskanje poti do zastave . . . . .	9
3.3 Prenos zastave v domačo bazo . . . . .	10
3.4 Srečanje z drugimi agenti . . . . .	10
3.4.1 Srečanje z agenti nasprotne ekipe . . . . .	11
3.4.2 Žrtvovanje agenta . . . . .	12
<b>Seznam slik</b>	<b>13</b>
<b>Seznam tabel</b>	<b>13</b>

# 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<sup>A</sup>T<sub>E</sub>X.

## 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 domišljjsko okolje. Bodisi z neposrednim igranjem, bodisi z opisovanjem njihovih dejanj nato igrajo te vloge v zgodbi, ki je lahko vnaprej načrtana 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 opisuje količino življenja ob začetku igre.
Speed	Hitrost, ki opisuje za koliko kock se bojevnik lahko premakne ob premikih.
Energy	Začetna 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 bojevnikar 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
Tank	boljša	boljši	višja	slabše	boljša
Vojak	slabša	slabši	nižja	boljše	slabša

# 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.

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.

**Izrek 2.1.1** (Minimax izrek) *V vsaki igri za dva igralca z ničelno vsoto*

*Let  $G$  be a finite group, and let  $H$  be a subgroup of  $G$ . Then the order of  $H$  divides the order of  $G$ .*

#### 2.1.1 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.

## 2.2 Akcije agenta

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

### 2.2.1 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.

### 2.2.2 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.

### 2.2.3 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 3

# Problemi na katere smo naleteli

### 3.1 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).

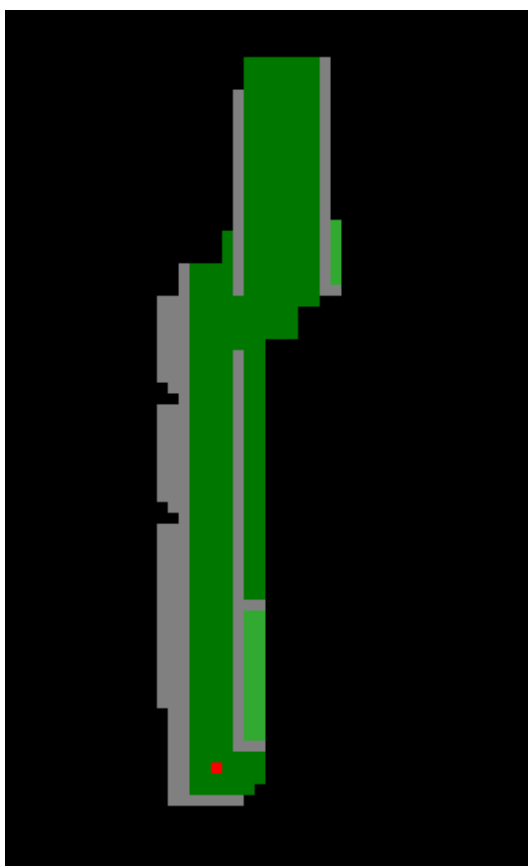
Raziskovanje je implementirano tako, da vse neobiskane celice oceni in nato izbere najboljšo. Ocena se izračuna na podlagi tega, koliko novih celic odkrijemo če obiščemo to pozicijo, ta ocena je skalirana z razdaljo te pozicije od trenutne pozicije. Majhen vpliv na oceno pa predstavlja bližina zidu. 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 heuristiko 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.

## 3.2 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 3.1 so z svetlejšo barvo označeni predeli, do katerih s trenutnim znanjem agent ne zna priti.



Slika 3.1: 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.

### 3.3 Prenos zastave v domačo bazo

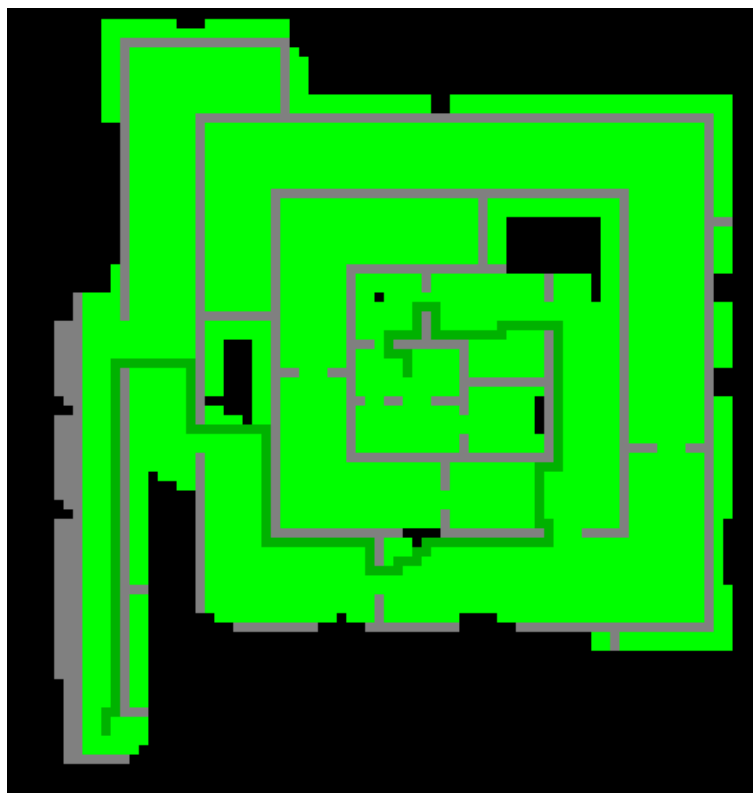
Ko agent pobere zastavo je njegova naloga, da jo prenese v bazo svoje skupine. Pri tem pa lahko naleti na agenta nasprotne ekipe, ki ga po vsej verjetnosti poizkuša ovirati na njegovi poti. Agentu ne preostane drugega, kakor da se mu umika. Vendar pa je gibanje agenta, ki nosi zastavo počasnejše od gibanja 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 3.2 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.

### 3.4 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 3.2: Agentov plan od zastave do baze

### 3.4.1 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.

### 3.4.2 Žrtvovanje agenta

Agenta ob trku oba izgubita življenje. Agent, ki izgubi življenje izgub 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

3.1	Agentova predstavitev sveta . . . . .	9
3.2	Izdelan plan . . . . .	11

# Tabele

1.1	Akcije bojevnikov . . . . .	4
1.2	Atributi bojevnikov . . . . .	4
1.3	Značilnosti bojevnikov . . . . .	5