**SVEUČILIŠTE U SPLITU**

**PRIRODOSLOVNO-MATEMATIČKI FAKULTET**

Seminarski rad

Oggy i žohari

Ivana Rogulj

Split, 20. veljače 2018.

Sadržaj

[1 Zadatak i cilj 1](#_Toc506857686)

[2 Svijet 2](#_Toc506857687)

[3 Agenti 3](#_Toc506857688)

[4 Pretrage 6](#_Toc506857689)

[4.1 Pretraga po dubini 7](#_Toc506857690)

[4.2 Pretraga po širini 8](#_Toc506857691)

[4.3 Pohlepna pretraga 9](#_Toc506857692)

[4.4 Pretraga A\* 10](#_Toc506857693)

[4.5 Dijkstra 11](#_Toc506857694)

[5 Usporedba algoritama 12](#_Toc506857695)

[6 Zaključak 18](#_Toc506857696)

# Zadatak i cilj

Projekt je nazvan Oggy i žohari po poznatom crtiću. Cilj ove igrice je da Oggy uspije pobjeći od neprijatelja i skupi sve kuglice. U Net-logo simulaciji nalazi se labirint sa žuto-rozim poljima i 4 vrste agenata od čega su 3 žohara, 1 pas Jack predstavljeni kao neprijatelji, mačak Oggy kao mačak te kuglice i zvijezde koje Oggy jede.  
Moguće je izabrati da se svi neprijatelji kreću jednom vrstom pretrage ili da se svatko kreće zasebnom pretragom dok se Oggy uvijek kreće jednom pretragom kojom pokušava pobjeći od neprijatelja.



# Svijet

Zadano je diskretno okruženje, labirint je prikazan na koordinatnom sustavu. Labirint je otvorenog tipa iako je moguće samo na jednom mjestu prijeći s jednog ruba svijeta na drugi. Svijet predstavlja mreža polja, odnosno podijeljen je na 10 x 10 kvadratića zvanih patchevi. Patchevi žute boje predstavljaju prazan prostor po kojem se moguće kretati, a patchevi roze boje predstavljaju zid.

Zadani svijet se postavlja pritiskom na botun Nova igra. Izgled samog labirinta se učitava pomoću metode import-pcolors. Sva su polja označena i to na način da su na labelama patcheva žute boje ispisane vrijednosti x \* 100 + y, gdje x i y predstavljaju koordinate svakog patch-a.

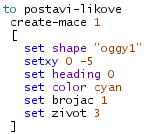
****

Brojčane oznake ne predstavljaju stvarnu udaljenost između dva polja, ali ih koristimo kako bismo onemogućili agentima da se kreću poljima koje su već posjetili te kao glavne elemente nekih pretraga.

# Agenti

Postoje četiri vrste agenata u prije opisanom svijetu.

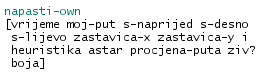
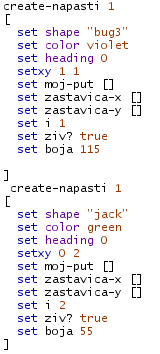
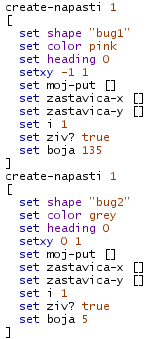
Prvu skupinu agenata čini mačak Oggy - agent koji se kreće po prostoru i koji pokušava pobjeći od neprijatelja.

1.PNG

Mačku sam dodijelila dodatno svojstvo:

Život – može imati 3 života i pomoću ovog svojstva se prati koliko mu je još života ostalo.

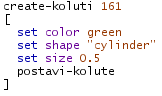
Drugu skupinu agenata čine neprijatelji takozvane napasti – agenti koji se kreću po prostoru i koji pokušavaju uhvatiti mačka.



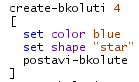
Neprijateljima sam dodijelila dodatna svojstva:

1. Vrijeme – ovo svojstvo služi kako bi pratili koliko vremena će neprijatelji biti ugroženi i u mogućnosti da ih Oggy pojede. Nakon što Oggy pojede plavu zvijezdu, ona će mu omogućiti da pojede neprijatelje.
2. Moj-put – ovo svojstvo služi kako bi mogla pratiti kojim putem se neprijatelji kreću kako se ne bi neprestano kretali istim putevima.
3. S-naprijed, s-desno te s-lijevo – ova svojstva služe kako bi provjerila koja su polja oko neprijatelja slobodna za kretanje po labirintu.
4. Zastavica-x te zastavica-y – ovo su dvije liste koje služe kako bi zapisala x i y koordinate patcheva na kojima dolazi do mogućnosti odabira neprijatelja u kojem smjeru će se kretati.
5. i – ova varijabla mi povećava razine kod pretrage po širini
6. Heuristika – koristi se kod pretrage A\* i računa veličinu mjesta na kojem se nalazi neprijatelj
7. Astar – se koristi kod pretrage A\* i zbraja vrijednosti varijabli heuristika i procjene-puta
8. Procjena-puta – se koristi kod pretrage A\* i računa veličinu mjesta na kojem se nalazi Oggy
9. Ziv? – se koristi kako bi se pretraga mogla zaustaviti. Ako nijedan od neprijatelja više nije živ, pretraga se zaustavlja.
10. Boja – svojstvo koje pamti početne boje neprijatelja. Služi kako bi se neprijatelji vratili u početno stanje nakon što istekne vrijeme u kojima ih Oggy može pojesti.

Treću skupinu agenata čine koluti.   
Oggy mora pojesti sve kolute kako bi pobijedio neprijatelje i pri tome skuplja bodove.



Četvrtu skupinu čine bkoluti odnosno zvijezde.   
Kada Oggy pojede plavu zvijezdu, neprijatelji postaju plave boje na određeno vrijeme u kojem ih može pojesti i tako smanjiti broj neprijatelja.



8.PNG

Patchevi posjeduju neka svojstva koja omogućuju praćenje kretanja agenata.

1. Brojač – ovo svojstvo koristim u postavljanju koluta u svijet. Kada je brojač jednak nuli tada će se na tim patchevima moći postaviti zeleni koluti. Na mjesta na kojima je brojač jednak 1 postavljaju se zvijezde.
2. Raskrižje? – ovo bool svojstvo koristim kod pretraga kako bih obilježila gdje mi dolazi do mogućnosti više izbora smjera neprijatelja.

Postoje još i globalne varijable.

10.PNG

1. Lista – ova lista služi kod A\* pretrage i dijkstre. Ona sprema određene vrijednosti potrebne za računanje funkcija ovih dvaju algoritama.
2. x i y – varijable koje pamte koordinate Oggya i također se koristi kod pretrage A\* te dijkstra.
3. Vrijeme1 – varijabla koja služi kao brojac vremena izvođenja pretraga
4. Bodovi – varijabla koja broji koliko je bodova skupio Oggy jedeći kuglice

# Pretrage

Detaljno ću objasniti pretrage koje sam implementirala u projektu. U cilju je prikazati kojim algoritmom najbrže dođe do kraja simulacije bilo to da je Oggy pojeo sve kuglice ili su neprijatelji pojeli Oggya..

Pretraživanja se dijele na slijepa i usmjerena. Razlika je što kod slijepog pretraživanja nemamo unaprijed nikakve informacije za problem koji rješavamo, odnosno nemamo znanje o putu koji je potrebno prijeći i ne znamo koliko je agent udaljen od cilja dok usmjerena pretraživanja koriste heurističke funkcije i tako stječu neka saznanja o cilju i udaljenosti.

Pretrage koje sam koristila su pretrage temeljene na pravim pretragama s tim imenom. Naravno, malo su prilagođene okruženju te labirintu. Nije bilo moguće sto posto prikazati kako neki algoritmi stvarno funkcioniraju, kao što je to moguće prikazati u stablu ili grafu, ali implementirane pretrage poštuju glavne uvjete algoritama te njihovu učinkovitost.

Uvjeti koji su postavljeni za sve pretrage:

Pretraga staje kada Oggy ostane bez života ili kad Oggy pojede sve kolutiće. Oggy se kreće po posebnoj pretrazi kojom pokušava pobjeći od neprijatelja, ali i skupiti što više bodova, dok se neprijatelji kreću po već zadanim pretragama, odnosno njenim pravilima.

Korištenjem svojstva moj-put, pamti se put kojim neprijatelji prolaze i tako se spriječava prelazak istim putem.

## Pretraga po dubini

Pretraga po dubini je algoritam pretrage koji potencijalna rješenja obilazi „grananjem u dubinu“.

Navedena pretraga djeluje na sljedeći način: na polju na kojem se nalazi neprijatelj učitaju se slobodna polja za kretanje i u ako ima više slobodnih polja neprijatelj će uvijek izabrati lijevo polje ako je slobodno. Postupak se nastavlja sve dok ne dođe do mjesta gdje ne može više ići ni u jednom smjeru ili dok ne dođe do kraja pretrage. Kada dođe do mjesta gdje ne može više ići nigdje vraća se na posljednji čvor gdje se nalaze još neispitani putevi.   
Svojstvo moj-put pamti put kojim se neprijatelji kreću tako s čime se onemogućava da se neprijatelji vrte u krug. Odnosno kada se neprijatelj vraća na neistražene puteve, pomoću tog svojstva se onemogućava da ponovno izabere isti put.

Nedostatak navedene pretrage je velika prostorna kompleksnost.

## Pretraga po širini

Pretraga po širini je algoritam koji ispituje sve čvorove na određenoj razini prije prelaska na slijedeću razinu. Potpuno je, što znači da uvijek pronalazi rješenje i implementira se tako što se izvršava širenje korijena.

Navedena pretraga djeluje na sljedeći način: neprijatelj učita slobodna polja za kretanje pa obilazi sve čvorove koji se nalaze na početnoj razini. Nakon što su svi čvorovi na toj razini istraženi dopušteno mu je istražiti sve čvorove na idućoj razini i tako sve dok ne dođe do posljednje razine. Nakon što se poveća razina neprijatelj se vraća na neki od neistraženih polja na jednu razinu niže kako bi istražio sve moguće čvorove na idućoj razini.

Razlika između prave pretrage po širini i prilagođene je itekako vidljiva. Prava pretraga po širini, ide do kraja, tj. dok svaki dio stabla ne bude istražen, ili u ovom slučaju dok se ne dogodi da je Oggy ostao bez života ili da je Oggy pojeo sve kolutiće.

Svojstvo moj-put pameti sve čvorove koji su posjećeni kako se neprijatelj ne bi stalno vrtio u krug i išao istim putem. To je ujedno i jedan od nedostataka jer se sama pretraga teško implementira kao prava jednoagentska pretraga po širini.

## Pohlepna pretraga

Najpoznatiji primjer usmjerenog pretraživanja je pohlepna pretraga. Ona je zapravo podvrsta usmjerenog pretraživanja, odnosno heuristička – koristi heurističku evaluacijsku funkciju.

Navedena pretraga djeluje na sljedeći način: kada neprijatelj učita koja su polja slobodna za kretanje, usporedi vrijednosti patcheva (plabel od svakog slobodnog polja) te ide na onaj patch čija je vrijednost najmanja. Pamti mjesta na kojima dolazi do više mogućnosti, pa ako ne dođe do cilja neprijatelji se vraćaju na čvorove u kojima je moguće ići još kojim izborom puta i uvijek izabire onaj koji ima najmanju vrijednost.

Optimalnost rješenja ovisi o strukturi svijeta što zapravo znači da može pronaći jako loša rješenja. U ovom labirintu zbog puta i prepreka, stvara se nepotrebno putovanje.

Pohlepna pretraga najbolja je kad se radi o praznom prostoru (ili onom sa što manje prepreka).

## Pretraga A\*

A\* algoritam pripada algoritmima usmjerenog pretraživanja te postupcima optimalnog pretraživanja. Ovaj algoritam bira sljedeći čvor koji će proširiti, ne samo na osnovu cijene njegove putanje, nego i na osnovu procijenjene udaljenosti do cilja odnosno heurističke funkcije.

Navedena pretraga djeluje na sljedeći način: prvo se definira lista u koju se sprema vrijednost heurističke funkcije (heuristika + procjena-puta), a nakon toga se poziva funkcija koja računa vrijednosti polja na kojem se nalazi neprijatelj (heuristika) i vrijednost polja na kojem se nalazi Oggy (procjena-puta). Uz to se poziva metoda koja služi da bi se neprijatelji kretali po labirintu.

## Dijkstra

Dijkstrin [algoritam](https://hr.wikipedia.org/wiki/Algoritam), služi za nalaženje najkraćeg puta u usmjerenom [grafu](https://hr.wikipedia.org/wiki/Graf_(struktura_podataka)) s nenegativnim vrijednostima rubova.

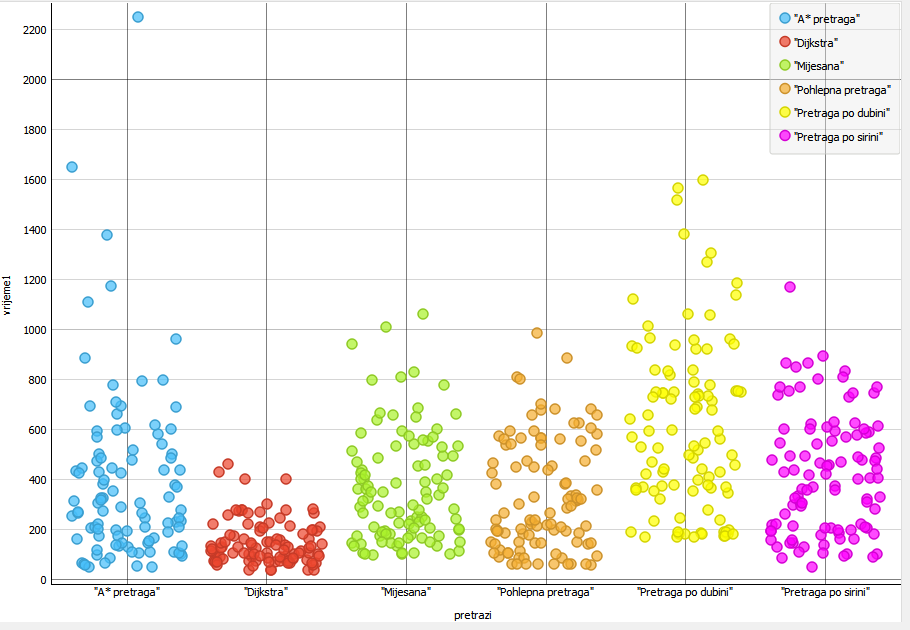
Na primjer, ako [čvorove](https://hr.wikipedia.org/wiki/%C4%8Cvor) predstavimo kao [gradove](https://hr.wikipedia.org/wiki/Grad), a vrijednosti rubova kao udaljenosti između onih gradova koji su direktno povezani, Dijkstrin algoritam nalazi najkraći put između dva grada.

Navedena pretraga djeluje na sljedeći način: definira se lista u koju se sprema vrijednost heuristike, odnosno vrijednost polja na kojem se neprijatelj nalazi, te se potom se poziva metoda koja služi da bi se neprijatelji kretali po labirintu.

# Usporedba algoritama

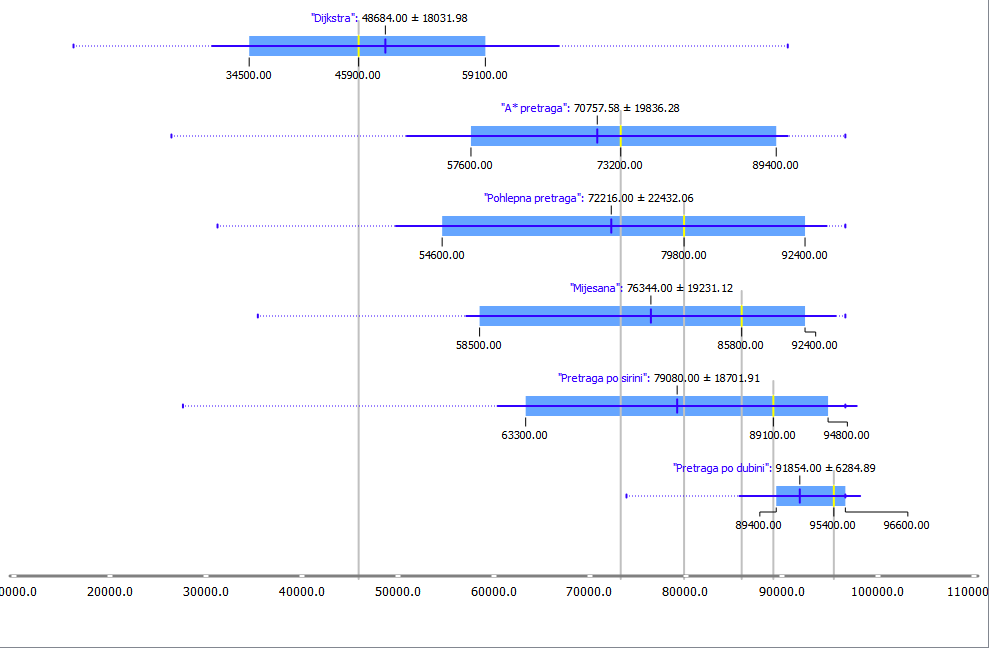
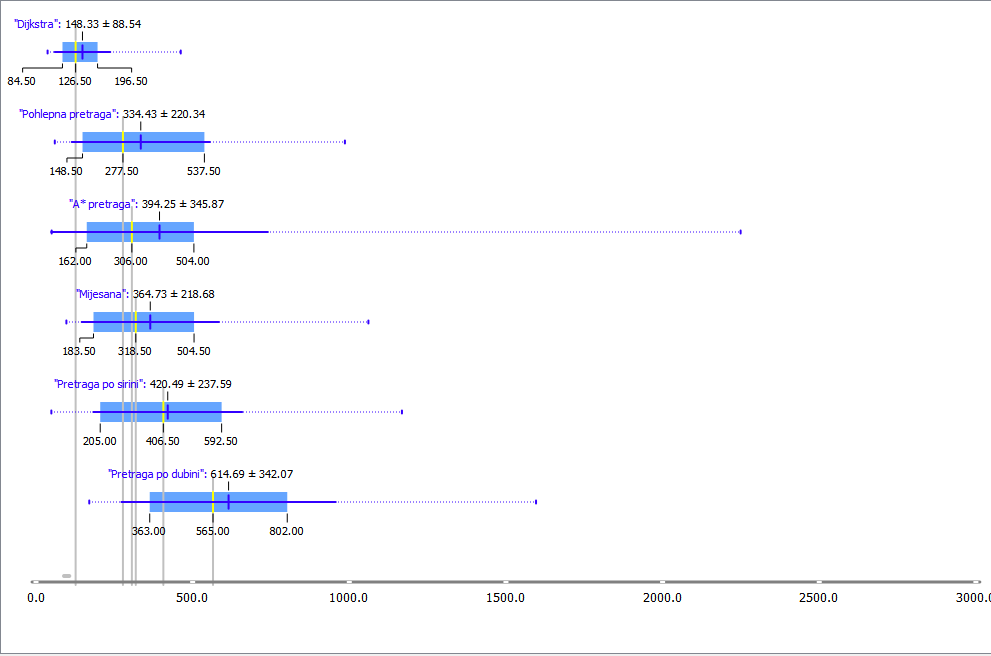
U projektu je implementirano šest različitih algoritama pretrage. Cilj u ovom projektu bio je s kojim se algoritmom dolazi najbrže do cilja igrice. Bilo to da je neprijatelj uspio pojesti Oggya ili je Oggy stigao pojesti sve kuglice te usporediti navedene algoritme kako je prikazano u sljedećoj tablici.

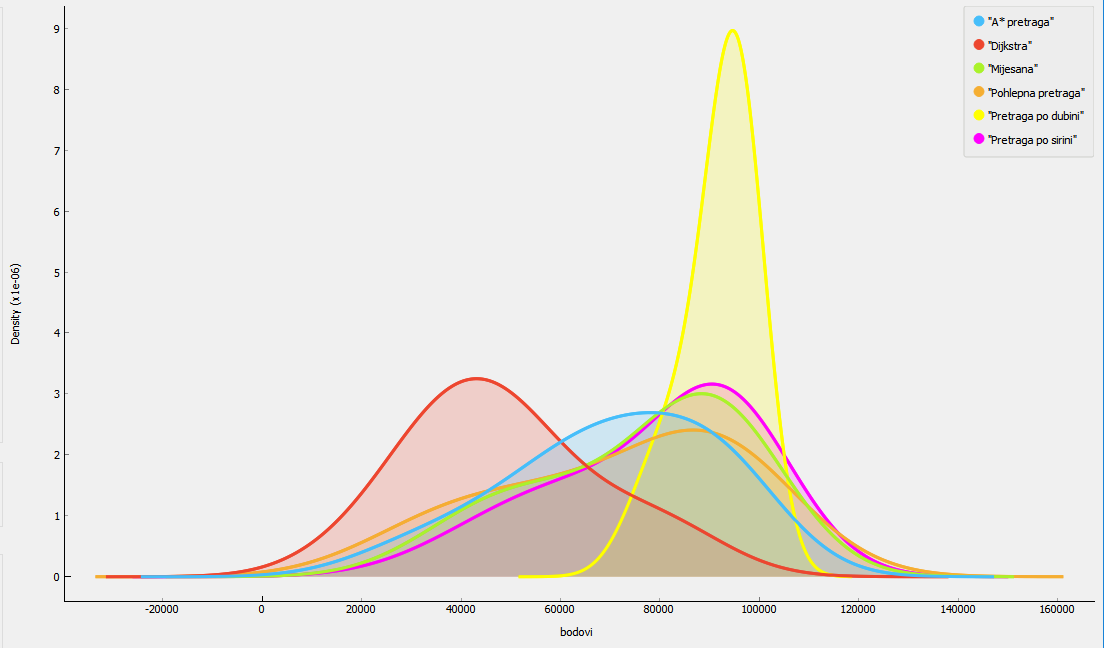
Za usporedbu algoritama sam koristila Behavior space te sam nakon njega podatke provukla kroz program Orange iz kojeg sam dobila statističke podatke u odnosu na bodove i vrijeme trajanja svakog algoritma.

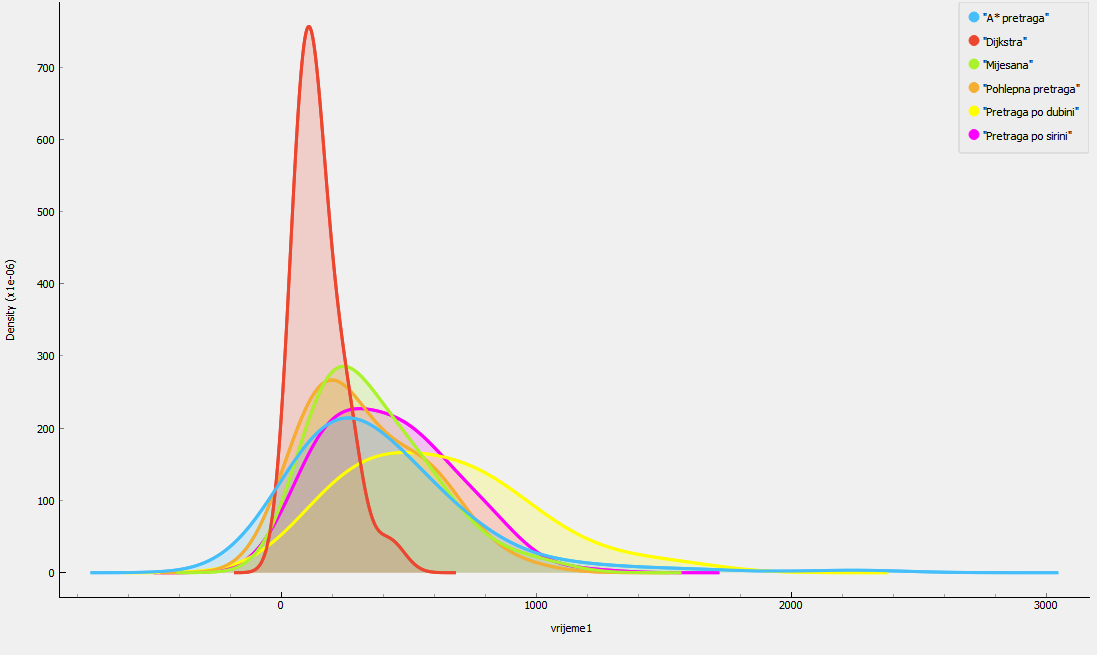


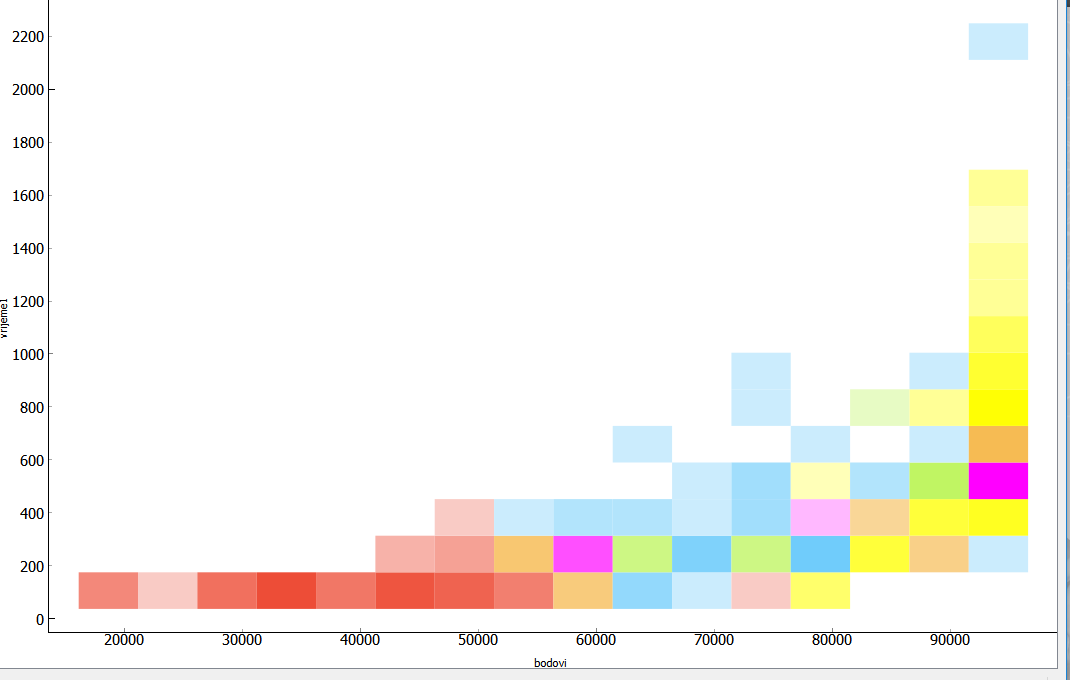
Najviše vremena za doći do cilja je bilo kada su se neprijatelji kretali pretragom po širini. A najmanje vremena je trebalo kada su se neprijatelji kretali dijkstrom.  
 

Dok se neprijatelji kreću pretragom po dubini Oggy je uspio pojesti najviše kuglica, a najmanje koristeći dijkstru.









# Zaključak

Cilj projekta bio je prikazati sve stečeno znanje ovog kolegija. Zaključila sam da svaka napravljena promjena na nekom algoritmu ima svoje prednosti i mane. Ali, i da ne postoji savršen algoritam pretrage.

Također, koristila sam jednoagentski sustav, što je mnogo otežalo implementaciju svakog algoritma. Takav je sustav u ovim pretragama doveo do veće potrošnje prostora i vremena no što je to općenito bilo potrebno.

Veliki zaključak cijelog projekta, ali i kolegija, je da izrada inteligentnog sustava nije jednostavna stvar te zahtijeva razumijevanje i umijeće rada s različitim algoritmima pretrage.

