**Agent za navigaciju dronova**

Marko Mihajlović 14742

Formulacija problema

Dinamika svakodnevnog života je u većim gradovima dovela je do pojave velikih gužvi u saobraćaju. Stvarajući se uglavnom obavljajući svakodnevnih aktivnoti, kao što je odlazak do prodavnice za namirnice.

Rešenje problema

Napraviti intaligentni dron koji će vršiti dostavu namirnica, od prodavnice (u daljem tekstu lokacija A) do naručioca (u daljem tekstu lokacija B). Nakon dostave potrebno je da se dron vrati na početnu lokaciju.

Opis problema:

* Inicijalno stanje (početna lokacija - lokacija A) vršilac usluge želi da dostavi kutiju do narucioca.  
  Unosi se lokacija na kojoj dron treba da dostavi paket korisniku. (lokacija B)
* Nakon isporuke potrbno je da se vrati do lokacije A

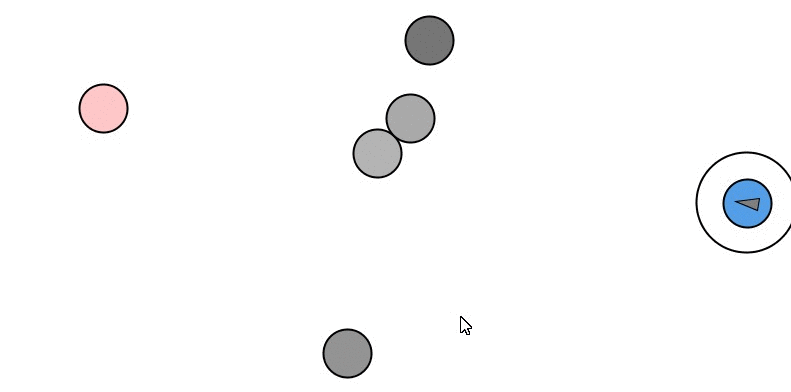
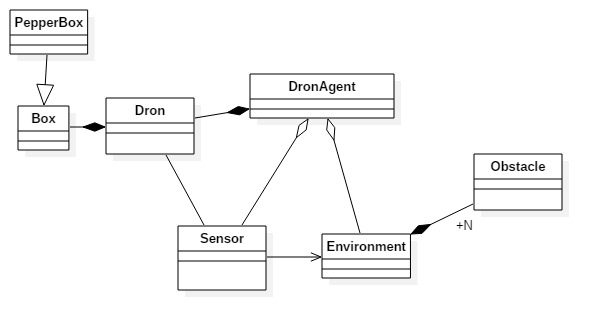
Nepredvidljive okolnosti:

* Prepreke izmedju zadatih destinacija

Potencijalno rešenje:

* Ugradnja senzora za detekciju i zaobilaženje prepreka.

Prva verzija ovog rešenja se zasniva na jednostavnom reflex agentu koji pamti stanja.



*[screenshot aplikacije]*

Implementacija rešenja:

1. Stanja Drona:

* Wait
* Running

1. Cilj – bezbedna dostava paketa do ružicaste tačke i povratak nazad
2. Providni krug predstavlja rang senzora, inicijalno je postavljen na 100px
3. Sive tačke na sredini terena predstavljaju prepreke koje je potrebno zaobići

Strategija:

1. Kada senzor detektuje prepreku obavestiće drona da se nalazi na nebezbednoj lokaciji i da treba da promeni svoju putanju. Senzor će proslediti odgovarajuću novu privremenu bezbedu lokaciju do koje treba da stigne.
2. Logika za odabir nove bezbedne lokacije:

* Uvek izabrati x koordinatu koja je ista primarnom cilju
* Koristiti stanje koje u zavisnosti od visine terena i stanja bazirnom na prethodnom promašaju vraća odgovarajuću vrednost y koordinate. Privremeni cilje je predstavljen praznim krugom.
* Isečak koda za odabir y koordinate privremenog cilja  
  do

{

switch (state) {

case 1: h = 0; break;

case 2: h = height;

case 3: h = height/4;

case 4: h = height/2; break;

case 5: h = 3 \* height/4; break;

case 7: h = height/3; break;

case 8: h = 2 \* height/3; break;

default:

h = random (0, height);

}

state = (state + 1) % 10;

p = new PVector (destination.x, h);

} while (! isSafe(p));

1. Kada Dron primi novu – privremenu lokaciju promeniće kursor ka toj lokaciji i pored toga će uložiti dodatan napor sabiranjem vektora određene jačine (*escapeFactor* atribut klase Drone). U zavisnosti od jačine *escapeFactor*–a dron će promeniti pravac različitim intezitetom. Milnimalna vrednost je 0.3 kako ne bi došlo do sudara sa preprekom. Za veliku vrednost kao što je 33 dron će dobiti suprotan pravac kretanja.
2. Dron poseduje brojač *backoff* koji odbrojava iteracije praćenja privremog odredišta. Nakon isteka se prati primarna destinacija.

Strategija koja može poboljšati preformanse:

1. Ugraditi senzor koji će učiti na osnovu prethodnih promašaja
2. Odmogućiti da senzor dinamički podešava *escapeFactor i backoff*
3. Implementirati exponential backoff algorithm za *backoff*
4. Odredjivanje idealne osetljivosti senzora

*Source code, izvršna verzija aplikacije i gif aplikacije sa lošom kvalitetom se nalazi u prilogu.*