**Tehnička škola Čakovec**

**ELABORAT ZAVRŠNOG RADA**

**AI SIMULACIJA**

**Mentor: Učenik:**

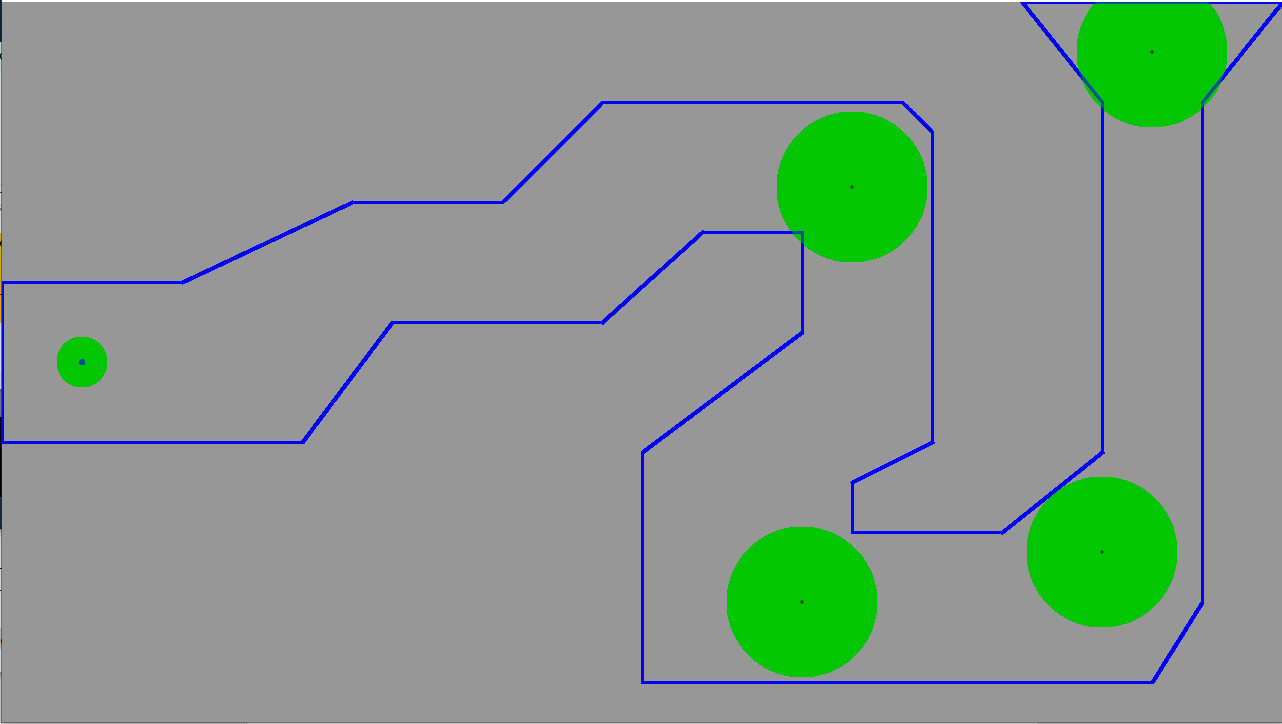
**Krešimir Kočiš, prof. Toni Polanec, 4.RT**

**Čakovec, svibanj 2020.**

Prva faza – kreiranje mape

Zbog samog funkcioniranja programa Processing način kreiranje staze za autiće morao je biti manualan. Kod pokretanja Processing programa kreira se kanvas za crtanje, te svi entiteti koje vidimo o programu moraju biti „nacrtani“. Tako i granice staze.

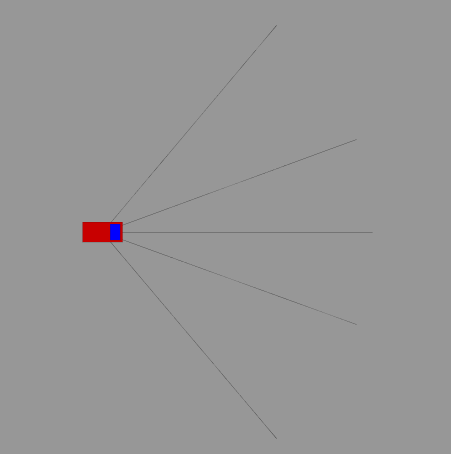
Jedna od bitnijih dijelova mape su takozvani „checkpointevi“ koji će nam služiti u genetskom algoritmu kroz koji ćemo proći kasnije. Oni su postavljeni na relativno bitne dijelove mape, zavoji ili teže prepreke. Kad auto uđe u radijus checkpoint-a tada program prepozna da je auto stigao do određenog dijela staze.



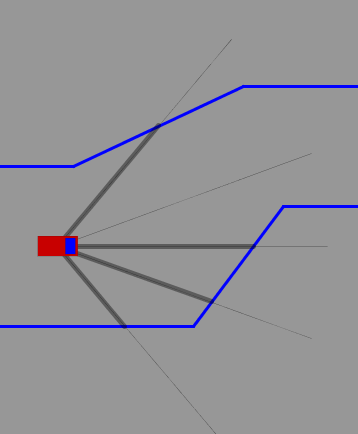
Druga faza – kreiranje autića

Tijelo auta je najobičniji pravokutnik te radi prepoznavanja koji je prednji i zadnji dio auta dodao sam prednje vjetrobransko staklo.

Najbitniji dio auta nije tijelo auta već senzori. Senzori nam prikazuju što i do koje udaljenost autić *vidi*. Senzori su postavljeni da *vide* 300 piksela ispred autića u 5 smjerova. Od kojih je jedan postavljen da gleda ravno naprijed, dva lijevo i desno pod 20 stupnjeva i dva lijevo i desno pod 50 stupnjeva od središnjeg senzora.



Da bi mi uočili i razumjeli što auto vidi, postavio sam uvjet ako senzor nešto prepozna ispred sebe, da se sama crta koja označuje senzor podeblja. S tim svojstvom nama je puno lakše vizualizirati autićevo vidno polje.



Problemi i rješenja

Problem 1: Auto ne detektira zid na vrijeme

Razlog: Funkcija isDead() gleda samo jedan senzor te po njegovoj vrijednosti postavlja dead na true ili false. Ako je taj određeni senzor okrenut lijevo, auto ne detektira zid s desne strane na vrijeme.

Rješenje: For petljom utvrđujem najmanju udaljenost zida svih senzora te istu šaljem isDead() funkciji.

Postupak generiranja nove populacije

1. Provjeravanje da li su svi autići završili u zidu
2. Izračunavanje ukupnog fitnessa svih autića
3. Postavljanje fitnessa svakog od autića po udjelu u ukupnom fitnessu

mojFitness = mojFitness / ukupniFitness;

1. Biranje 2/3 od ukupne populacije autiće koji će nastaviti putovanje pomoću Tournament selection.

Algoritam:

* + 1. Od svih autića biramo 2 autića nasumično.
    2. Od tih 2 uzmemo onaj s boljim fitnessom. Te ga zapisujemo u *winner* polje
    3. Ponavljamo korak 1 i 2 tako dugo dok ne dobijemo 2/3 od ukupnog broja autića u *winner* polje

1. Autići iz *winner* polja su odabrani za produciranje novih autića. (parent – child odnos). Child autić dobijemo tako da uzmemo 2 autića nasumično iz *winner* polja te radimo „crossover“. Pola neural network child dobije od prvog roditelja a drugu polovicu od drugog roditelja.
2. Mutation. Kao i u prirodi kod evolucije se događaju mutacije pa tako i u evolucijskome programu. Stopa mutacije je određena konstantom [0.0001, 0.1]. Stopa mutacije određuje koja je vjerojatnost da se određeni dio neural networka, tj. pojedini weight, promjeni.
3. Takvi novi autići prelaze u novu populaciju (1/3) zajedno sa svojim roditeljima (2/3).

Problem: Generacija 2 odmah mrtva, i neprestano generiranje novih generacija, također mrtvih. Debugiranje trajalo 4 sata. Rješenje: timer koji je mjerio vrijeme od početka programa ako auto ne prijeđe određeni checkpoint na vrijeme, umire se nakon završetka jedne populacije nije resetirao pa svaka sljedeća populacija i svi autići nisi bili u mogućnosti doći do checkpoint-a na vrijeme pa zato i svi bili mrtvi.