

Genetski algoritam za automatsko šutiranje u fudbalskoj igri

Uvod

Ideja za projekat je nastala dok sam radio na igri Golden Boot. Kako je igra namenjena za vise igraca koji istovremeno igraju, nastala je potreba za laznim igracima (botovima) koji bi zamenili prave igrace u slucaju kada nemamo dovoljno ljudi online da uparimo sve. Zadatak algoritma bio bi da za tacno definisanu tacku na голу uspe da nadje prave parametre i da sutne loptu, tako da ona sa velikom tacnoscu pogadja zadatu tacku. Igrac sutira loptu tako sto prstom prevlaci preko ekrana i kada predje odredjenu udaljenost, ispali loptu u pravcu u kom je povlacio prst. To znaci da su nam za sut potrebna 3 parametra koje algoritam treba da izracuna da bi sutnuo loptu:

- X i Y komponente vektora pravca koji formiramo povlacenjem prsta preko ekrana (vektor koji dobijamo izmedju pocetne i krajnje tacke na ekranu) - u nastavku deltaX i deltaY
- Vreme koliko smo povlacili prst jer to odredjuje jacinu suta - u nastavku deltaT

Genetski algoritmi su se cinili kao dobra opcija prvenstveno zbog toga sto se mehanika igre cesto menja pa nam ovaj pristup resavanju problema omogucava da jednom napisano resenje moze da se prilagodi vecini stvari koje izmenimo. Na primer algoritmu nije vazno da li zivi zid postoji ili ne, da li je golman na голу, da li postoji jak vetar i slicno.

Mana genetskog algoritma je sto nije dovoljno performantan na losijim mobilnim uredjajima bez velike optimizacije fizike koju trenutno koristimo u igri.

Sa genetskim algoritmima sam se prvo upoznao na kursu Vestacka inteligencija, ali sam to znanje hteo da prosirim i na video igre. Clanak od kog sam krenuo i kojim sam se vodio kroz razvijanje resenja se nalazi na ovom linku

http://www.csd.uwo.ca/~katchab/pubs/gameonna2008_unreal.pdf

Za resavanje problema sam se odlucio za programski jezik C# i razvojno orkuzenje Unity.

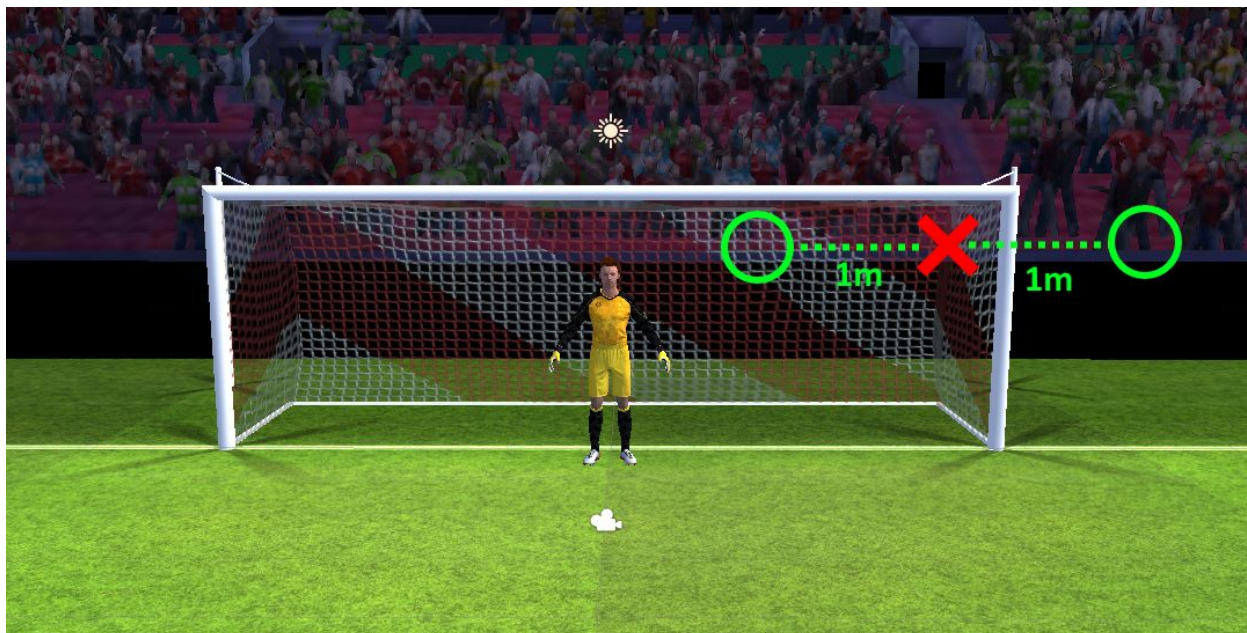
Opis resenja

Odmah na pocetku bilo je jasno da ce postojati 2 problema a to su nacin razmenjivanja DNK pri generisanju novih generacija i funkcija prilagodjenosti.

Za potrebe kodiranja 3 parametra deltaX, deltaY i deltaT napravljena je klasa DNA koja sadrzi ta 3 polja.

Problem koji sam prvo krenuo da resavam je bio funkcija prilagodjenosti. Prvo logicno resenje koje mi je palo na pamet je bilo da u trenutku kada lopta predje gol liniju (bilo da je pogodjen gol ili ne) racunam udaljenost lopte od zadate tacke na голу, medjutim odmah su se javila 2 problema. Prvo je sto podjednako kaznjavam i promasen i pogodjen gol iako je pozeljnije da

algoritam daje gol (ne bi trebalo da bude isto ako sam promasio za metar i dao gol, ili promasio za metar i pogodio van gola). Drugi problem je sto je moguće pogoditi stativu a onda lopta nikada ni ne predje gol ili aut liniju.



U ovom primeru je algoritmu receno da pokuša da pogodi gornji desni ugao, a on je podjednako kaznjavao i slučaj kad lopta udje u gol i pogodi van njega

Za sledeću iteraciju sam se odlucio da kaznjam promasaje vise nego pogotke tako sto cu mnoziti udaljenost sa nekim koeficijentom vecim od 1 sto je popravilo rezultate medjutim nedovoljno. Za suteve iz ostrih uglova se previse cesto desavalo da lopta pogadja stativu umesto gol, a za uglove kao na slici iznad bilo je dosta neprecizno gde lopta pogadja iako je uvek ulazila u gol (pa se desavalo da golman uspe da odbrani iako pokušavam da pogodim sam cosak)

Za poslednju iteraciju koja je pokazala najbolje rezultate implementirao sam detekciju sudara lopte i stative i te sudare kaznjavao sa udaljenost * 5, a promasaj van gola sa udaljenost * 20. Takva funkcija prilagodjenosti je davala odlicne rezultate i za suteve iz ostrih uglova.

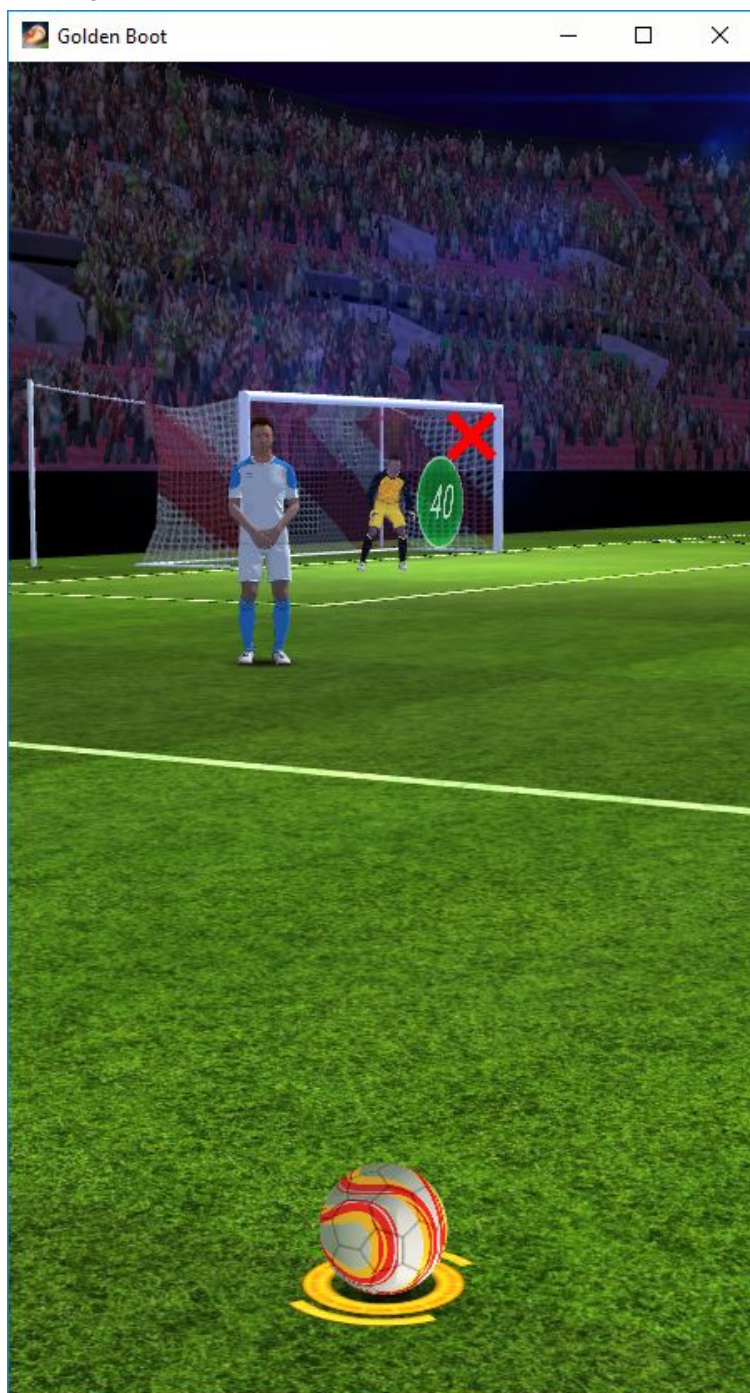
Drugi veliki problem na koji sam naisao bilo je ukrstanje 2 bota pri generisanju nove generacije. Ideja je bila da kada simuliram sve suteve u okviru jedne generacije, uzmem trecinu sa najboljim rezultatom i onda od njih generisem celu novu generaciju. Uzimao sam po 2 bota i od njih pravio 6. Pri prvoj iteraciji sam samo uzimao i mesao njihove parametre. Na primer jedan od novih botova je imao deltaX od prvog bota iz prethodne generacije, a deltaY i deltaT od drugog i obrnuto.

Takav pristup nije davao zadovoljavajuće rezultate jer su losiji botovi previse uticali na generisanje novih jedinki pa je konvergencija ka resenju predugo trajala. Nakon toga sam favorizovao boljeg bota od uzetog para tako sto sam njegove parametre vise ubacivao u novu generaciju nego od losijeg bota. To je donelo znacajnu razliku i algoritam je za samo 20 generacija poceo da pogadja izuzetno uspesno.

Sto se mutacije tice, tu sam nasumicno neki od parametara za neku malu vrednost povecavao ili smanjivao.

Eksperimentalni rezultati

Eksperimenti su vršeni tako sto je zadata tacka bila gornji desni ugao gola a onda je izvršeno po 10 suteva sa 3 razlicite pozicije na terenu. Korisceno je 24 bota po generaciji i 10 generacija ukupno sa stopom mutacije od 50%



Prva iteracija gde je funkcija prilagodjenosti podjednako kaznjavala i pogotke i promasaje i gde se ukrstanje vrsilo ravnomerno izmedju 2 bota:

	Pogodjeno	Promaseno
Sut iz levog ostrom ugla	2	8
Sut iz desnog ostrom ugla	3	7
Sut sa sredine terena	9	1

Druga iteracija gde je funkcija prilagodjenosti razlicito kaznjavala promasaje i pogotke, ali nije razlikovala pogodak stativne a ukrstanje je favorizovalo boljeg bota u paru:

	Pogodjeno	Promaseno
Sut iz levog ostrom ugla	6	4
Sut iz desnog ostrom ugla	7	3
Sut sa sredine terena	10	0

Poslednja i trenutno aktuelna iteracija je ona gde je implementirana i detekcija pogotka u stativu kao i favorizovanje boljeg bota pri ukrstanju:

	Pogodjeno	Promaseno
Sut iz levog ostrom ugla	10	0
Sut iz desnog ostrom ugla	10	0
Sut sa sredine terena	10	0

Kako je poslednja iteracija donela u potpunosti zadovoljavajuce rezultate, broj generacija sam smanjio na 20 i time popravio performanse za oko 50%, s'tim sto se preciznost malo pogorsala ali zanemarljivo. Sutevi iz ostrom ugla su poceli vise da pogadjaju stativne medjutim i dalje su lopte ulazile u gol.

Zakljucak

Pored toga sto je algoritam dao apsolutno zadovoljavajuće rezultate sto se tice preciznosti, pokazala se i jos jedna dobra strana. Naime, kako algoritam vise vrednuje pogodjen gol, u slucaju kada na terenu postoji zivi zid koji blokira tacku koju algoritam pokusa da pogodi, on ce odabrati takav sut tako da svejedno daje gol i pored zivog zida iako nece biti u tacku koja mu je zadata. Tu se takodje javlja ideja za unapredjenje algoritma a to je dodavanje parametara za zakrivljenost lopte tako da sut cak moze i da zaobidje zivi zid i pogodi zadatu tacku.

Najveci problem koji i dalje postoji je koriscenje fizike koja nije deterministicka tako da za svaki sut ja zapravo simuliram ceo sistem fizike na svakih 0.04 sekunde i onda proveravam ishod suta. Ogromna optimizacija bi bila da mogu na analiticki nacin da iz parametara suta odredim tacku koju ce lopta pogoditi na голу bez simulacije celog sistema.

Literatura

http://www.csd.uwo.ca/~katchab/pubs/gameonna2008_unreal.pdf - Osnovni koncepti genetskih algoritama za video igre

<https://docs.unity3d.com/Manual/index.html> - Dokumentacija Unity-ja koja mi je pomogla da preuzmem kontrolu nad sistemom fizike da bih mogao da simuliram suteve