

Računalniška igra z uporabo genetskega algoritma za učenje nasprotnikov

[Projekt pri predmetu Povezljivi sistemi in inteligentne storitve]

Luka Horvat

Univerza v Mariboru

Fakulteta za elektrotehniko, računalništvo in informatiko

Smetanova ulica 17, SI-2000 Maribor, Slovenija

luka.horvat@student.um.si

POVZETEK

Digitalne igre so dandanes vedno večji rastoči trg. Pojavljajo se na računalniških platformah, konzolah in prenosnih napravah. Del problema načrtovanja iger je implementacija dobre in zanimive umetne inteligence, zato v tem članku predstavim rešitev z uporabo genetskih algoritmov in uporabljeno okolje razvoja, predstavim rezultate in v zaključku podam svoje mnenje o teh tehnologijah.

Splošni pojmi

Theory Games AI Experimentation

Ključne besede

Unity, genetski algoritmi, igra

1. UVOD

Umetna inteligenca je po definiciji inteligenca, s katero razpolagajo naprave oziroma stroji, da bi posnemali človeško razmišljanje [6]. Računalniška igra pa je računalniški program, katerega glavni namen je razvedrilo in zabava. Dandanes se umetna inteligenca uporablja za nadzor različnih entitet v računalniških igrah, poraja pa se vprašanje ali je možno za to uporabiti genetske algoritme.

V članku o botih v igrah so N. Cole, S. J. Louis in C. Miles uporabili računalniško igro Counter Strike in nastavljali parametre računalniškim nasprotnikom. S pomočjo genetskega algoritma so dosegli enake rezultate, kot ko so bili parametri nastavljeni s pomočjo ekspertov. V drugem članku so s končnim avtomatom, ki se je spreminjal s pomočjo genetskega algoritma dosegli, da je umetna inteligenca zmožna preigrati različne stopnje v igri Infinite Mario Bros. V zadnjem članku pa so uporabili genetski algoritem za adaptacijo umetne inteligence, ki je igrala namizno igro Othello. [1, 3, 4]

V tem članku sem uporabil metodo, ki temelji na metodi,

opisani v [1]. Najprej sem ustvaril računalniško igro z nasprotniki, ki jih je možno upravljati s pomočjo parametrov. Te sem nato nastavljal z genetskim algoritmom.

Članek je organiziran v naslednje odseke. V odseku 2 podrobno opišem problem in predstavim implementacijo v odseku 3. Sledijo rezultati v odseku 4 ter zaključek v 5.

2. OPIS PROBLEMA

Glavna problema mojega dela sta implementacija same računalniške igre in nato uporaba genetskega algoritma za nastavljanje parametrov nasprotnikov.

2.1 Računalniška igra

Osnovna ideja moje računalniške igre je preprosta 2D vesoljska strelska igra. Igralec kontrolira svoje vesoljsko plovilo, ki ga lahko premika po vertikalni in horizontalni osi. Pri tem je omejen na območje igralnega prostora. Obenem ima možnost streljati na nasprotnike, vendar mora paziti, saj lahko naenkrat obstaja samo en njegov izstrel v igri. Najpomembnejši del igre pa so nasprotniki. Približno enake velikosti kot igralec se v valovih vključujejo v igro na vrhu igralnega prostora. Nakar pa se glede na nastavljene parametre premikajo po polju in izvajajo vrsto akcij. Njihov cilj je uničiti igralca s pomočjo svojih izstrelkov ali pa s kolizijo z njim. Parametri nasprotnikov so:

- Začetna smer premika, ki definira kam se bo nasprotnik ob vstopu v polje začel premikati.
- Štirje smerni parametri, ki definirajo kam se premakniti ko dosežemo rob igralnega polja.
- Vektor umika od igralčevega izstrelka.
- Seznam akcij, ki jih periodično izvajajo:
 - Naključen premik v levo ali desno smer.
 - Premik v stran od igralca.
 - Premik proti igralcu s poskusom kolizije z njim.
 - Streljanje projektilov proti igralcu.

Igralec ima na voljo pet življenj in poskuša preživeti čim več valov nasprotnikov, s čimer napreduje po stopnjah. Potrebno se je izmikati kolizijam z nasprotniki in jih obenem hitro uničiti z izstrelki.

2.2 Genetski algoritem

Genetski algoritem je heuristika, ki uporablja procese naravne selekcije in evolucije ter se velikokrat uporablja za iskanje optimizacij in rešitve za iskalne probleme. Glavni koraki algoritma so naslednji [5]:

1. Definicija sestave rešitve problema. Rešitev je predstavljena z vrsto parametrov ali tako imenovanih kromosomov. Z njimi je možno oceniti kvaliteto rešitve in tako posamezne rešitve urediti po uspešnosti.
2. Izgradnja začetne populacije rešitev. Število je odvisno od problema, so pa večinoma ustvarjena naključno, da imamo širok spekter rešitev.
3. Posameznike ocenimo in naredimo selekcijo za naslednjo iteracijo. Metod izbire je več:
 - Rulet selekcija - posameznikova možnost selekcije je sorazmerna z normalizirano oceno.
 - Turnir selekcija - izbranih je n naključnih posameznikov, med katerimi pa izberemo najboljšega. Postopek ponovimo k krat.
 - Krajšanje - izberemo polovico najboljših posameznikov.
4. Na izbranih posameznikih izvedemo genetske operacije:
 - Križanje - dva posameznika sta izbrana in iz njiju ustvarimo sinova, ki imata križane genetske zapise staršev. Metod je več, v osnovni varianti pa izberemo mesto križanja in zlepimo levo polovico prvega starša z drugo polovico drugega starša.
 - Mutacije - ustvarjenim otrokom z določeno verjetnostjo vnesemo majhne spremembe v genih. Tako se izognemo lokalnim optimumom v iskanju rešitve.
5. Ponavljamo postopek od 3. točke dalje. Zaključimo, ko preteče določeno število iteracij, potekel čas ali smo našli dovolj dobro rešitev.

S temi postopki pridemo do zelenih rešitev za izbran problem.

3. IMPLEMETACIJA REŠITVE

Za implementacijo igre sem uporabil *Unity Engine*. Je igralni pogon, s pomočjo katerega lahko z uporabo grafičnega vmesnika in programskega jezika C# izdelamo 2D ali 3D računalniško igro in jo brez sprememb izvozimo za različne platforme kot so Windows, Linux, OS X, Android, iOS, WebGL ipd. Pogon je na voljo zastoj za manjše komercialne projekte in vsebuje zelo obsežno dokumentacijo [2]. Za izdelavo grafičnih elementov sem uporabil grafični vektorski urejevalnik *Inkscape*, nekatere elemente pa sem pridobil s spleta z ustrezno licenco.

Glavni del igre sestavlja igralčeva ladja, ki jo igralec lahko premika s pomočjo smernih tipk ali z dotikom na zaslon na mobilni napravi. Z gumbom pa lahko izstrelil projektil. Za upravljanje z nasprotniki sem izdelal menedžerja. Ta skrbi za ustvarjanje novih nasprotnikov in poganja sam genetski algoritem. V vsaki generaciji je bilo 5 nasprotnih

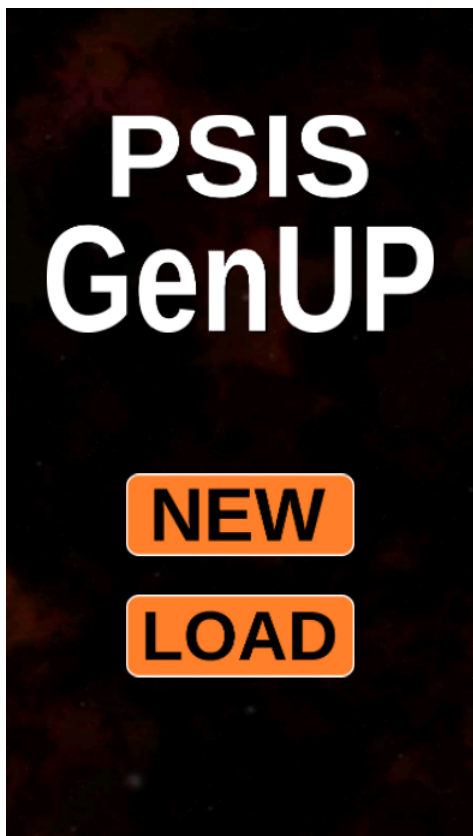


Slika 1: Glavni del igre. Igralec poskuša uničiti nasprotnike.

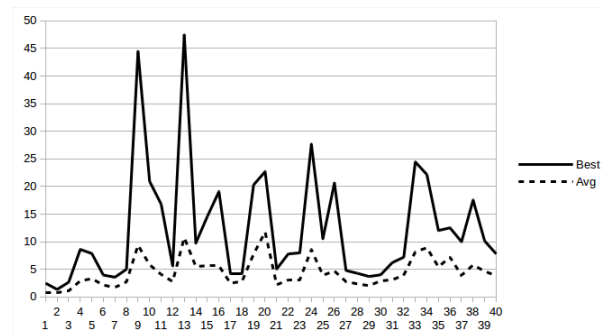
ladij. Parametre posameznikov sem v genu predstavil z binarnim zapisom. Parametri smeri so bili predstavljeni z osmimi bitmi, seznam akcij pa vsaka z dvema bitoma (možne so štiri akcije). Te gene sem nato posredoval posamezni nasprotni ladji, ki je izvela potrebne podatke. Ladje so se nato pojavile v igri in poskusile uničiti nasprotnika. Ocena ladje pa je bila odvisna od tega, koliko sekund je preživela v igri. Če je zadela igralca z izstrelkom, sem prištel 15 sekund, če pa se je sama zaletela v igralca, pa sem prištel 10 sekund. Ko je igralec uničil vse trenutne nasprotnike, sem jih razvrstil po vrsti glede na to, koliko sekund so preživeli. Nad posamezniki sem nato izvedel rulet selekcijo in nad otroci s 5% možnostjo izvedel mutacijo na posameznih bitih gena. Postopek sem ponavljal dokler igralec ni izgubil vseh življenj. Igro lahko vidimo na sliki 1.

Implementiral sem tudi glavni meni igre, kjer je možno izbrati novo igro ali nadaljevati prejšnjo. To je mišljeno v smislu ponovnega pogona genetskega algoritma. Pri novi igri ustvarimo novo naključno začetno populacijo, pri nadaljevanju pa nadaljujemo kjer smo končali zadnjič. Rezultat lahko vidimo na sliki 2.

Sami igri sem dodal tudi glasbeno podlago, shranjeval sem stanje genetskega algoritma za večkratno igranje in aplikacijo podprl tudi za igranje na mobilnih napravah s pomočjo zaslona na dotik.



Slika 2: Glavni meni igre.



Slika 3: Najboljša in povprečna ocena posameznikov.

4. POSKUSI IN REZULTATI

Za potrebe analize sem osebno igral igro ter meril najboljšo in povprečno oceno posameznikov preko 40 generacij. Rezultat vidimo na sliki 3. Iz rezultatov lahko razberemo, da vrednosti zelo nihajo. Največji problem je subjektivni vpliv igralčeve sposobnosti. Tudi če algoritem najde dobro rešitev, bo dober igralec nasprotnike hitro uničil in tako zmanjšal ocene posameznikov. Odvisno je tudi, katere nasprotnike se odloči igralec uničiti najprej. Če uniči boljše najprej, potem se bo algoritem zmedel in bodo rezultati slabši. Če igralca izključimo iz algoritma, se nasprotniki zelo hitro naučijo preživeti v igralnem prostoru (če zapustijo igralno polje potem umrejo), vendar to ni bil cilj projekta in analize. Po mojem subjektivnem mnenju igra postaja vedno težja, vendar se igralec prilagodi in to nasprotnikovo prednost izniči.

Za pridobitev boljših rezultatov bi bilo zanimivo aplikacijo ponuditi drugim uporabnikom preko Play trgovine in narediti širšo analizo algoritma.

5. ZAKLJUČEK

Uspelo mi je implementirati zaključeno računalniško igro ter vključiti genetski algoritem za učenje in prilagajanje nasprotnikov. Glavni problem projekta je analiza uspešnosti algoritma, saj je odvisna od posameznega igralca. Kot navedeno v prejšnjem poglavju, bi bilo potrebno vključiti večjo množico ljudi in opraviti statistično analizo. Vseeno je osnova zelo zanimiva in je možnosti za nadaljnjo raziskavo veliko.

6. REFERENCES

- [1] N. Cole, S. J. Louis, and C. Miles. Using a genetic algorithm to tune first-person shooter bots. In *Evolutionary Computation, 2004. CEC2004. Congress on*, volume 1, pages 139–145. IEEE, 2004.
- [2] U. G. Engine. Unity game engine - official site. [Online][Cited: 15. Junij, 2016.] <http://unity3d.com>.
- [3] N. C. Hou, N. S. Hong, C. K. On, and J. Teo. Infinite mario boss ai using genetic algorithm. In *Sustainable Utilization and Development in Engineering and Technology (STUDENT), 2011 IEEE Conference on*, pages 85–89. IEEE, 2011.
- [4] K. Ikeda, Y. Tanaka, S. Viennot, N. H. Quoc, and Y. Ueda. Adaptation of game ais using genetic algorithm: Keeping variety and suitable strength. In *Soft Computing and Intelligent Systems (SCIS) and*

13th International Symposium on Advanced Intelligent Systems (ISIS), 2012 Joint 6th International Conference on, pages 945–951, Nov 2012.

- [5] M. Mitchell. *An introduction to genetic algorithms*. MIT press, 1998.
- [6] S. Russell, P. Norvig, and A. Intelligence. A modern approach. *Artificial Intelligence*. Prentice-Hall, Englewood Cliffs, 25:27, 1995.