

Jetpack FERride
Tehnička dokumentacija
Verzija 1.0

Studentski tim:

Ivan Mikulić
Mate Gašparini
Antun Magdić
Adi Čaušević
Martin Čekada
Matija Bertović

Nastavnik:

Marko Čupić

Sadržaj

1. Opis razvijenog proizvoda.....	1
2. Tehničke značajke.....	2
Unaprijedna neuronska mreža	2
Elmanova neuronska mreža	3
Genetsko programiranje.....	3
Evolucijski algoritam	4
3. Upute za korištenje.....	5
Pokretanje aplikacije.....	5
Igranje igre.....	5
Pokretanje računalnog igrača	6
Treniranje novog računalnog igrača	8

1. Opis razvijenog proizvoda

Jetpack FERride je projekt razvijen u sklopu „Projekt iz programske potpore“ na Fakultetu Eelektrotehnike i računarstva sveučilišta u Zagrebu, akademske godine 2018./2019.

Projekt je razvijen uz mentorstvo dr. sc. Marka Čupića, a projektni tim čine Ivan Mikulić, Mate Gašparini, Antun Magdić, Adi Čaušević, Matija Bertović i Marin Čekada.

Računalna igra bazirana je na mobilnoj igrici „Jetpack Joyride“.

U igri je cilj izbjegavati prepreke, skupiti što više novčića i preživjeti što je moguće dulje.

Prepreke se u igri generiraju beskonačno, dok se igračeva brzina postepeno povećava.

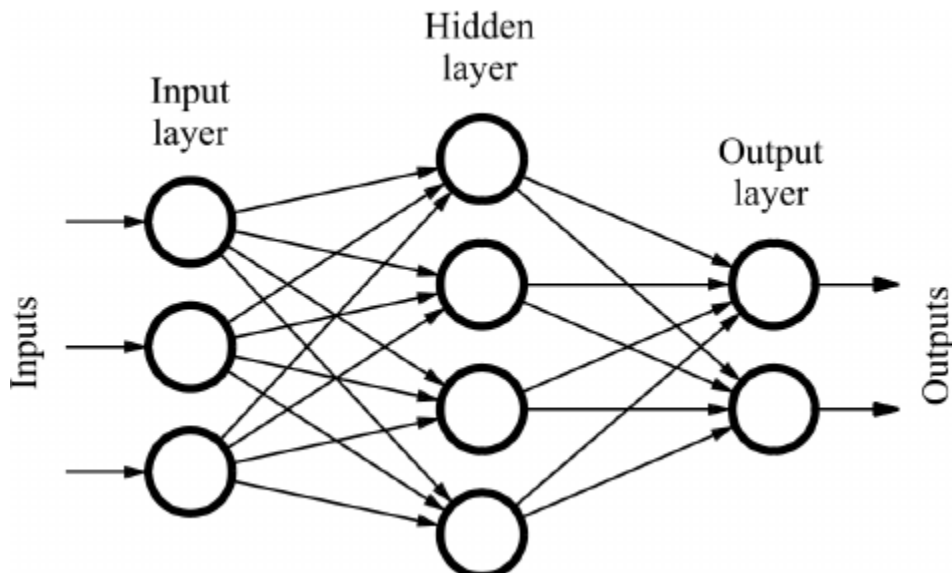
Primarni cilj projekta nije bio razvoj igre, nego pokazat sposobnost računala da uz pomoć specifičnih algoritama samo nauči igrati igru.

Program dolazi sa datotekama u kojima se nalaze od prije istrenirani autonomni igrači koje se može učitati i onda promatrati kako oni savladavaju problem.

Program također nudi mogućnost treniranja novog autonomnog igrača i pohranjivanje istog u datoteku. Postoje tri pristupa kojima se računalo može učiti, od korisnika se traži da prilikom pokretanja i treniranja na početku odabere koji od ta tri osnovna pristupa želi koristiti.

2. Tehničke značajke

Unaprijedna neuronska mreža



Slika 1 - dijagram unaprijedne neuronske mreže

U ulaznom sloju dobivaju se podaci o trenutnom stanju modela igre. Ti podaci odražavaju trenutno stanje igre i omogućuju računalu da ima potrebne podatke za donošenje odluke. Ti podaci se zatim množe sa težinama svake veze i na njih se još dodaje vrijednost otklona. Rezultat tog množenja i zbrajanja se zatim provlači kroz prijenosnu funkciju, u ovom slučaju sigmoidu i ta se vrijednost pohranjuje u neuronu prvog skrivenog sloja.

Postupak se provodi za svaki par neurona koji je povezan nekom vezom sve dok se ne izračuna vrijednosti svih neurona.

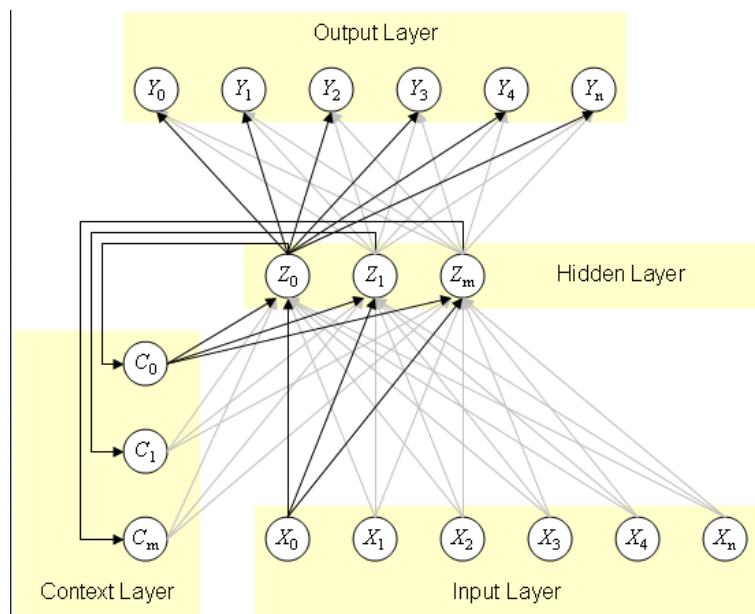
Nakon izračuna na temelju vrijednosti koje su pohranjene u neuronima izlaznog sloja donosi odluka o interakciji sa igrom.

Ulazni sloj se sastoji od 51 neurona. U njima su sadržani podaci kao što su kut pod kojim se vide rubovi prepreka, apsolutna vertikalna udaljenost do prepreke, brzina igrača, kutovi pod kojima se vide novčići itd. Svi ulazni neuroni skalirani su tako da im vrijednosti budu apsolutno manje od 1.

U našoj igri budući da je jedina interakcija korisnika sa igrom odluka o tome hoće li se model igrača uzdizati ili padati, izlazni sloj ima samo jedan neuron.

Prilikom treniranja od korisnika se traži da unese dimenzije skrivenih slojeva.

Elmanova neuronska mreža

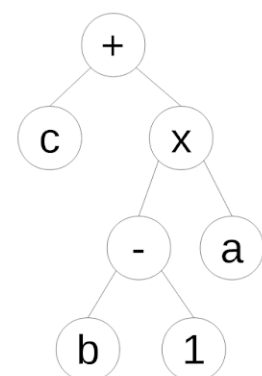


Slika 2 - dijagram Elmanove neuronske mreže

Elmanova neuronska mreža pripada porodici rekurzivnih neuronskih mreža. Razlika u odnosu na prethodno opisane unaprijedne neuronske mreže je u tome da postoji dodatni context layer. Ulazi u taj sloj su izlazi prvog skrivenog sloja, a taj sloj svoje izlaze ponovno šalje na prvi skriveni sloj, ali sa različitim težinama. Znači da kod Elmanove neuronske mreže prvi skriveni sloj ne prima samo ulaze modela, nego i izlaze iz prvog skrivenog sloja iz prethodne iteracije. Sve ostale funkcionalnosti mreže, kao što je izračun izlaza, ulazni podaci i slično, jednako je kao i kod prethodno opisane unaprijedne neuronske mreže.

Genetsko programiranje

Genetsko programiranje kao model "mozga" koristi stablo koje predstavlja program koji se može evaluirati da bi se dobila neka vrijednost na temelju koje se donosi odluka o akciji. Čvorovi 'a', 'b' i 'c' su ulazi programa, a vrijednost čvora '+' je izlaz. Ako je izlaz pozitivan jetpack će biti uključen, a ako nije pozitivan jetpack će biti isključen. Na početku učenja svi se čvorovi odaberu nasumično i nakon toga se genetskim algoritmom traže bolja stabla. Pri tome koriste se operacije križanja i mutacije. Prilikom operacije križanja na temelju dva stabla (roditelja) stvaraju se dva dijeteta tako da se roditeljima zamijeni jedno nasumično odabrano podstablo podstablom iz drugog roditelja. Prilikom mutacije jedno nasumično odabrano podstablo jedinice zamjenjuje se novim nasumično stvorenim podstablom.



Slika 3 - dijagram genetskog stabla

Evolucijski algoritam

Za samo učenje neovisno o odabranom pristupi koriste se evolucijski algoritmi.

Evolucijski algoritam sastoji se od 3 faze:

1. selekcije
2. križanja
3. mutacije

U fazi selekcije se prema nekom kriteriju biraju jedinke koje će sudjelovati u fazi križanja. Postoji mnogo načina za odabir tih jedinki i za svaki od tri osnovna pristupa implementirani su algoritmi koji po nekom parametru to rade i na korisniku je da prilikom treniranja odabere i podesi metodu koju želi koristiti.

U fazi križanja se na osnovu prethodno selektiranih jedinki stvara nova jedinka. Tu također postoji mnogo načina za stvaranje novog djeteta i implementirane su različite metode koje se mogu izabrati.

U fazi mutacije karakteristike novo stvorene jedinke imaju šansu da budu promijenjene. Ova faza postoji da bi se izbjeglo zapinjanje rješenja u lokalnim optimumima, tj. da bi se obogatio genetski materijal generacije. Za ovu fazu također postoje različite implementacije i na korisniku je da ih odabere.

3. Upute za korištenje

Pokretanje aplikacije

Za pokretanje aplikacije potrebno je skinut .jar datoteku sa web stranice i imati instaliranu Javu minimalno verzije 8.

Instalacijski paket Jave možete preuzeti na slijedećoj poveznici:

<https://www.java.com/en/download/>

Igranje igre

Na glavnom izborniku izabere se gumb „Play“ što pokreće igricu. Držanjem tipke „space“ na tipkovnici model igrača počinje ubrzavati ka vrhu ekrana, prestankom držanja tipke akceleracija modela igrača počinje opadati i nakon nekog vremena postane negativna te igrač počne padat.

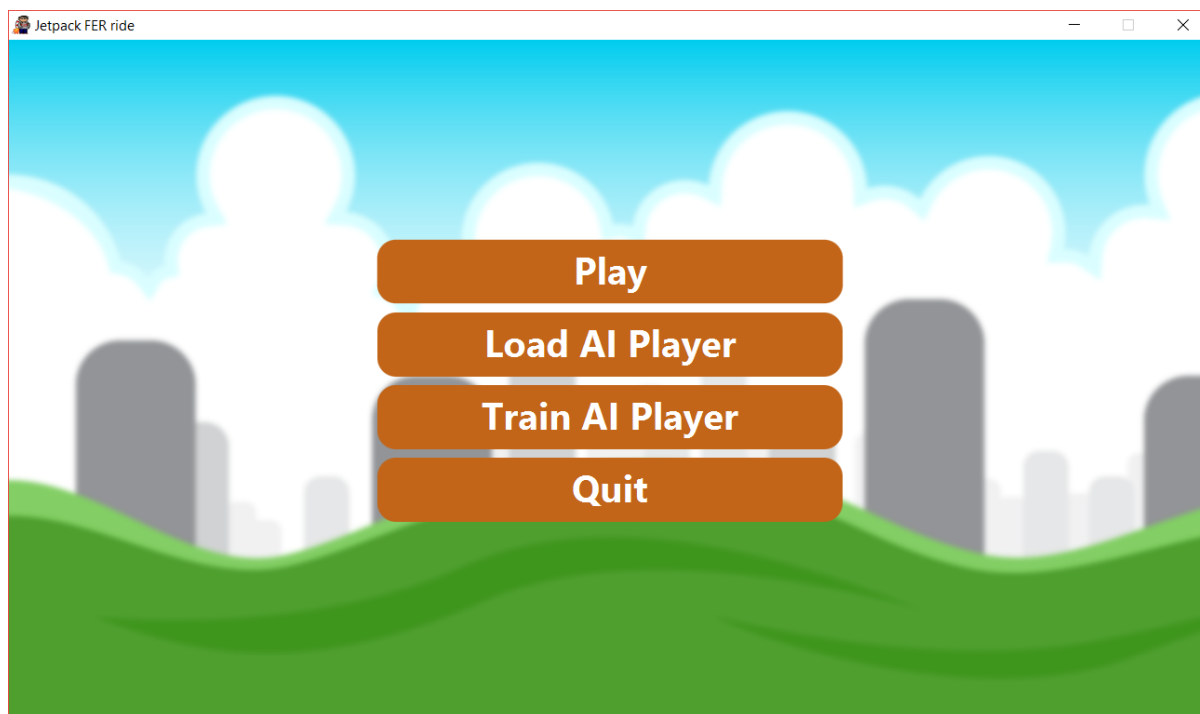
Postoje 3 vrste objekata sa kojima model igrača ima interakciju.

1. Prepreke
Prepreke mogu biti stacionarne ili pomičuće. Dodirne li model igrača u bilo kojem trenutku prepreku igra se zaustavlja
2. Novčići
Skupljanjem novčića igraču se povećava broj bodova koje je do sada skupio. Svaki novčić povećava broj bodova za 150. Novčić se skupi tako što ga model igrača dotakne.
3. Znanstvenici
Dodirne li model igrača znanstvenika igra se zaustavlja. Preleti li igrač prenisko iznad znanstvenika on ga „ubije“ i gubi 1000 bodova.

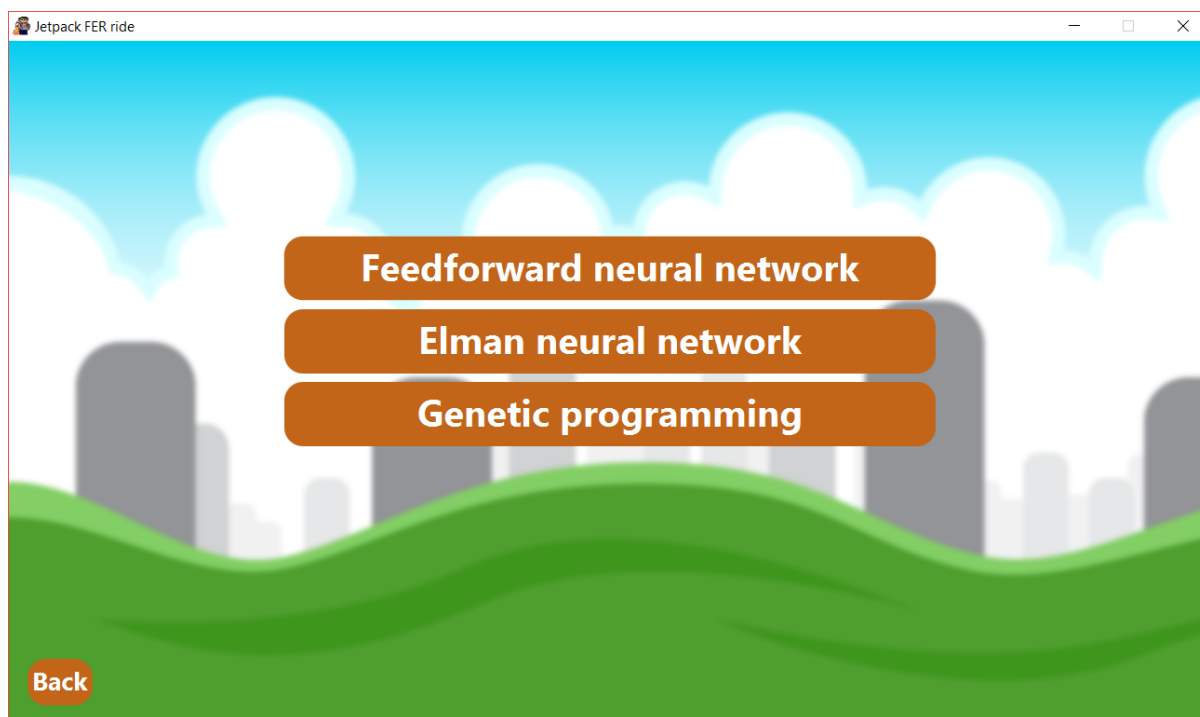
Na kraju igre nudi nam se dugmad koja omogućuje ponovo igranje ili povratak na glavni izbornik.

Pokretanje računalnog igrača

Na glavnom izborniku izabere se gumb „Load AI Player“, to nas odvodi na drugi izbornik na kojem biramo koji algoritam želimo da naše računalo koristi. Nakon odabira algoritma prikazuje nam se lista svih od prije istreniranih „igrača“.



Slika 4 - glavni izbornik



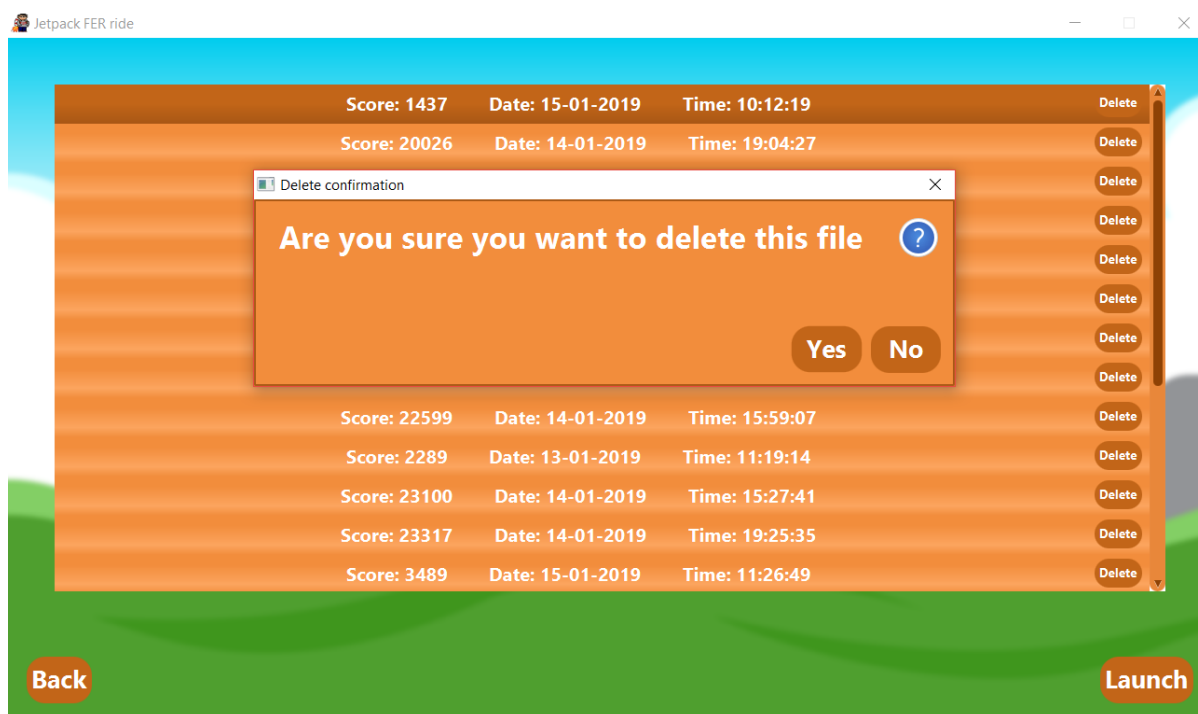
Slika 5 - odabir algoritma

Navedeni su bodovi koje igrač obično postiže te datum i vrijeme kada je stvoren. Pored svakog „igrača“ nalazi se i gumb „delete“ koji omogućuje brisanje igrača što dovodi do nepovratnog brisanja datoteke koja ima pohranjenog igrača.

Pritiskom na gumb „Launch“ pokreće se odabrani „igrač“ i vidimo na ekranu kako igra.



Slika 6 - izbornik za odabir igrača



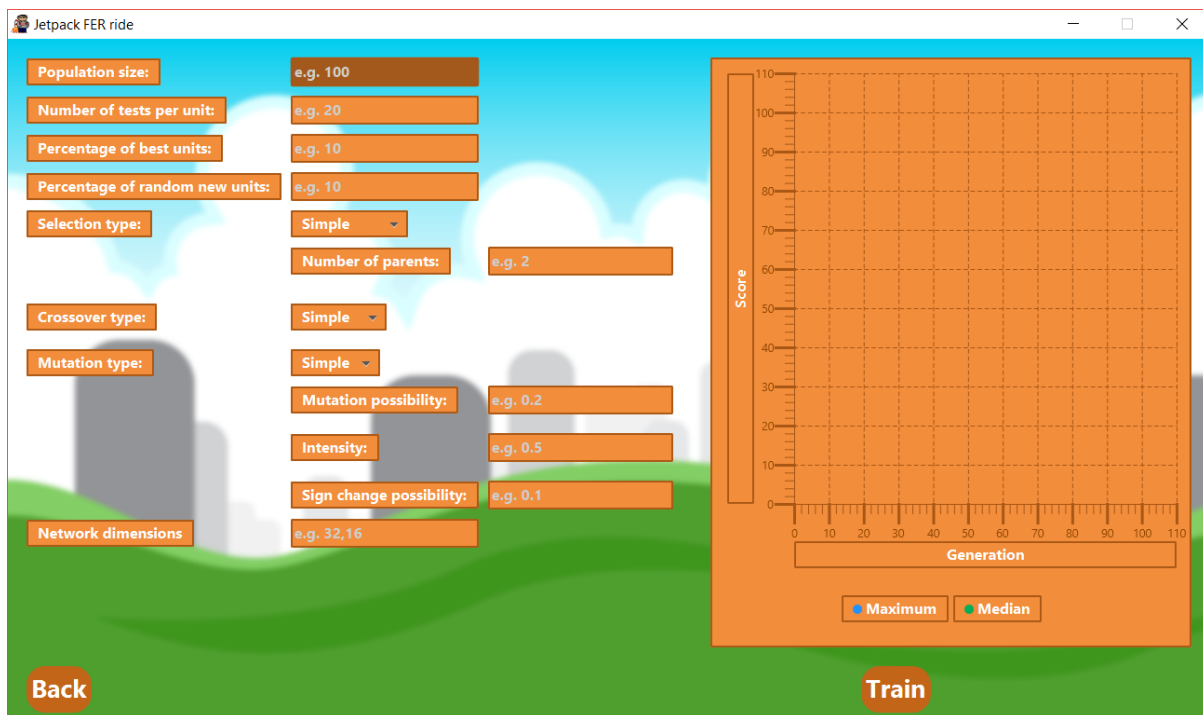
Slika 7 - prozor za potvrdu odluke o brisanju

Treniranje novog računalnog igrača

Na glavnom izborniku odabire se gumb „Train AI Player“. To nas odvodi na drugi izbornik gdje biraмо po kojem osnovnom algoritmu želimo trenirati. Pritiskom na gumb za algoritam koji želimo koristiti dolazimo na sučelje za treniranje. Odabire se koje verzije koraka evolucijskih algoritama želimo koristiti i unosi se parametre potrebne za te verzije. Ako iz imena parametra nije jasno čemu on služi, postavi se pokazivač miša iznad polja za unos teksta i pojavljuje nam se oblačić sa detaljima.

Pritiskom na gumb „Train“ pojavljuju započinje proces treniranja. Na grafu nam se prikazuju vrijednosti bodova trenutno najboljeg igrača u generaciji i vrijednost medijana bodova u cijeloj generaciji. Gumb „Train“ preimenuje se u „Stop training“ i sva polja za unos podataka od prije postaju neizmjenjiva.

Pritiskom na gumb „Stop training“ prekida se proces treniranja i sprema se „igrač“ sa trenutno najvećim brojem bodova. Taj se igrač zatim može pokrenuti kao i svi ostali do sada spremljeni „igrači“.



Slika 8 - izbornik za treniranje