

# **Učenje neuronske mreže evolucijskim algoritmom na primjeru igre – neslužbena dokumentacija**

**Studentski tim:**

Domagoj Latečki  
Boris Generalić  
Jure Čular  
Damir Kopljar  
Martin Matak

**Mentor:**

Marko Čupić

|  |                  |
|--|------------------|
| Učenje neuronske mreže evolucijskim algoritmima na primjeru igre | Verzija: 1.0     |
| Neslužbena dokumentacija   | Datum: 24.1.2016 |

## Sadržaj

|     |  |    |    |
|-----|--|----|----|
| 1.  | Opis razvijenog proizvoda                | 3  |    |
| 2.  | Tehničke značajke                        | 4  |    |
| 2.1 | Igra                                     | 4  |    |
| 2.2 | Neuronske mreže                          | 7  |    |
|     | <b>2.2.1 Potpuno povezana mreža</b>      |    | 7  |
| 2.3 | Evolucijski algoritmi                    | 10 |    |
|     | <b>2.3.1 Klasični genetski algoritam</b> |    | 10 |
| 2.4 | Ulazi u neuronsku mrežu                  | 10 |    |
| 3.  | Korištene tehnologije                    | 12 |    |
| 3.1 | Programski jezik Java                    | 12 |    |
| 3.2 | Tehnologija JavaFX                       | 12 |    |
| 4.  | Literatura                               | 13 |    |
| 5.  | Tablica sadržaja                         | 13 |    |
| 5.1 | Slike                                    | 13 |    |
| 5.2 | Tablice                                  | 13 |    |
| 5.3 | Algoritmi                                | 13 |    |

|  |                  |
|--|------------------|
| Učenje neuronske mreže evolucijskim algoritmima na primjeru igre | Verzija: 1.0     |
| Neslužbena dokumentacija   | Datum: 24.1.2016 |

## 1. Opis razvijenog proizvoda

Igra napravljena tijekom razrade projekta inspirirana je igrom „Flappy bird“. Cilj igre je izbjeći što više zapreka i prikupiti što više bodova. Prikupljanje bodova se vrši tako da igrač izbjegava zapreke i prikuplja nagrade. Kako bi igrač što dulje preživio ne smije izaći iz ograničenog prikaza slike te dotaknuti dno ili zapreke. Za svaku zapreku koju igrač uspije proći bez da ju dotakne igrač će biti nagrađen s jednim bodom, dok će za svaki skupljeni novčić igrač biti nagrađen s 10 bodova.

Igraču je omogućeno i uređivanje postavki unutar igre. Unutar postavki korisnik može podešavati razne parametre kao što su: gravitacija, širina cijevi, visina cijevi, vjerojatnost pojavljivanja novčića itd.

Osim same igre koju igrač može igrati, korisnik unutar izbornika može pokrenuti i igru koju će igrati naučena neuronska mreža.

Neuronska mreža koja reprezentira igrača naučena je genetskim algoritmom.

Kao selekcijski operator koristila se turnirska selekcija, za operator križanja korišteno je *BLX-alpha* križanje, dok je za mutaciju korištena Gauss-ova mutacija s definiranom vjerojatnošću.

Korištena neuronska mreža sadrži 9 ulaza, 50 čvorova unutar skrivenog sloja i 1 izlaz.

|  |                  |
|--|------------------|
| Učenje neuronske mreže evolucijskim algoritmima na primjeru igre | Verzija: 1.0     |
| Neslužbena dokumentacija   | Datum: 24.1.2016 |

## 2. Tehničke značajke

### 2.1 Kako funkcionira igra ?



Slika 2.1 Početni izbornik

Korisnik je u mogućnosti sam igrati igru ili pokrenuti razvijenog igrača da igra igru. Odabir načina igranja se izabire u početnom izborniku, gdje se pritiskom na tipku „**Play**“ pokreće igra u kojoj korisnik može sam igrati igru, a pritiskom na tipku „**Play AI**“ pokreće se igra kojom upravlja trenirana neuronska mreža na temelju danih ulaza. Pritiskom na tipku „**Settings**“ otvara se izbornik s postavkama, a pritiskom na tipku „**Quit**“ izlazi se iz igre.

Model svijeta kao i izbornik je napravljen u tehnologiji JavaFX, objekti koji se prikazuju za vrijeme igranja su:

- Igrač - u igri predstavljen kao ptica
- Zapreke - igri predstavljene kao vertikalno postavljene cijevi
- Nagrade - u igri predstavljene kao novčići
- Dno - predstavljeno grafičkim prikazom trave
- Broj bodova koje je igrač trenutno prikupio

|  |                  |
|--|------------------|
| Učenje neuronske mreže evolucijskim algoritmima na primjeru igre | Verzija: 1.0     |
| Neslužbena dokumentacija   | Datum: 24.1.2016 |



Slika 2.2 Model svijeta dok igra korisnik

Postoji još i izbornik koji se otvara približavanjem miša lijevom rubu ekrana, u njemu se nalazi tipka „Back“, pritiskom na nju korisnik se vraća na početni izbornik. Dodatno u slučaju kada igračem upravlja neuronska mreža u tom izborniku postoji mogućnost uključivanja prikaza objekta koji svijetli crveno u trenutku kada igrač pritisne tipku za skok te linija koje iscrtavaju neke od ulaza u neuronsku mrežu. Uz to postoji i klizač pomoću kojeg se može usporiti ili ubrzati iscrtavanje modela svijeta na ekranu.



Slika 2.3 Model svijeta dok igra neuronska mreža

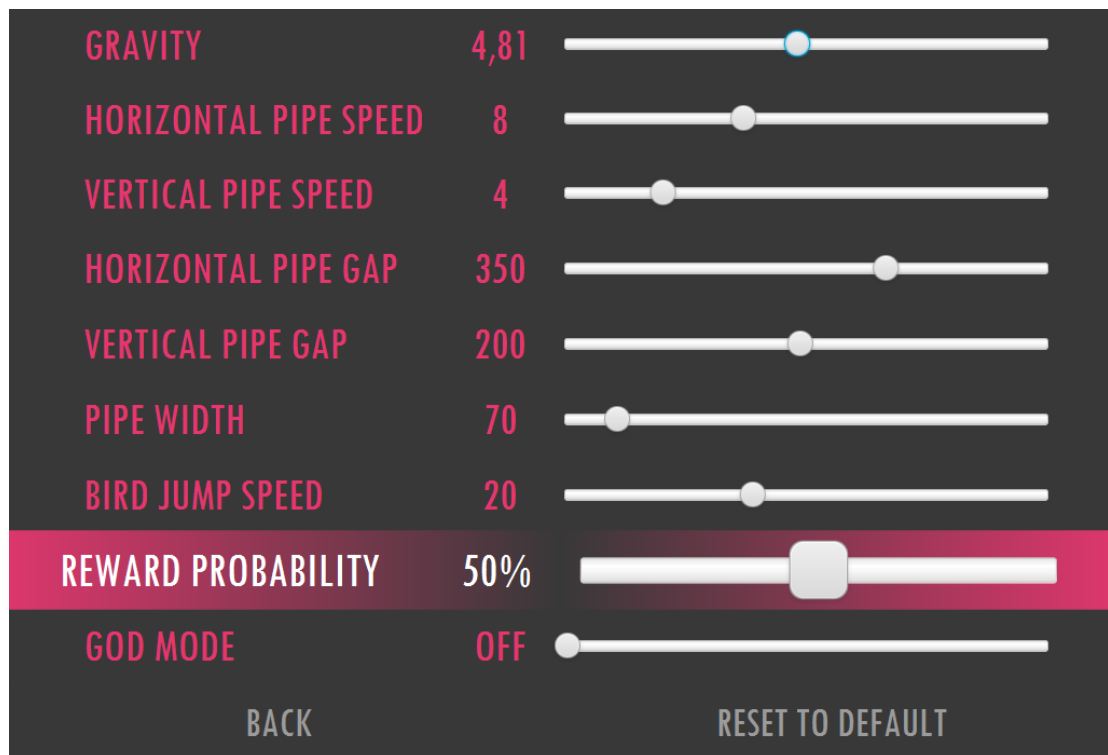
|  |                  |
|--|------------------|
| Učenje neuronske mreže evolucijskim algoritmima na primjeru igre | Verzija: 1.0     |
| Neslužbena dokumentacija   | Datum: 24.1.2016 |

Igrač objektom koji ga predstavlja upravlja pritiskom bilo koje tipke na tipkovnici osim tipke „B“. Pritiskom se objekt pomakne po vertikalnoj osi za predodređenu udaljenost(u daljnjem tekstu će se referirati da igrač „skoči“).

Za vrijeme igranja pritiskom na tipku „B“ se igra trenutno zaustavlja, a ponovnim pritiskom tipke „B“ se igra dalje nastavlja.

U izborniku za postavke - „Settings“, korisnik može postaviti razne opcije vezane uz model igre te postaviti uobičajene postavke. Moguće je mijenjati sljedeće postavke:

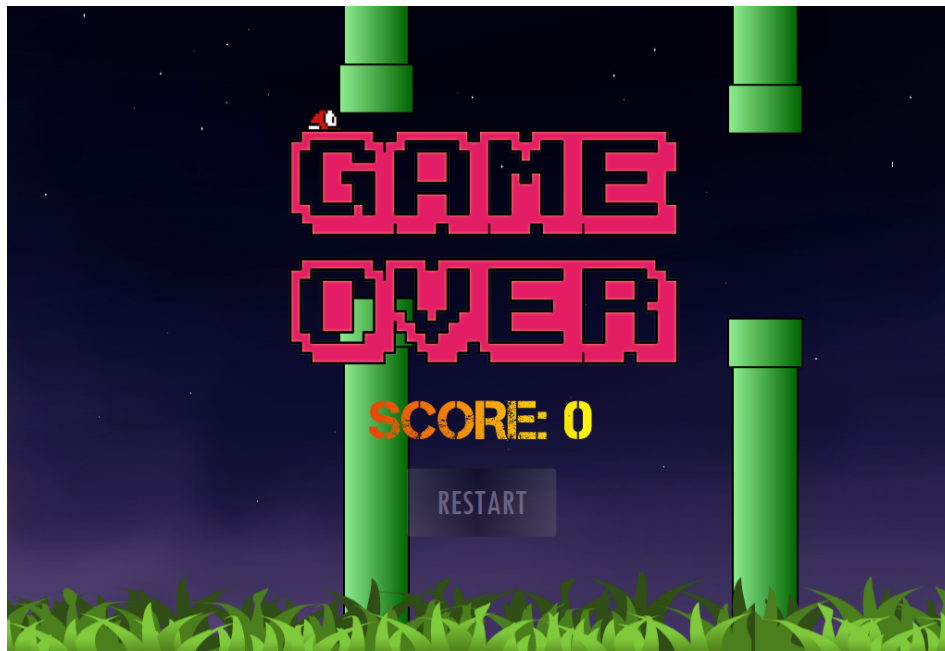
- Ubrzanje gravitacije
- Brzinu kojom se kreću zapreke po horizontalnoj osi
- Brzinu kojom se kreću zapreke po vertikalnoj osi
- Međusobnu udaljenost para zapreka
- Veličinu prolaza kroz svaku zapreku
- Širina zapreka
- Brzinom kojom igrač „skoči“
- Vjerojatnost pojavljivanja nagrada
- Opcija koja određuje može li igrač prolaziti kroz zapreke i ograničenja koja su postavljena u igri.



Slika 2.4 Izbornik postavki

|  |                  |
|--|------------------|
| Učenje neuronske mreže evolucijskim algoritmima na primjeru igre | Verzija: 1.0     |
| Neslužbena dokumentacija   | Datum: 24.1.2016 |

Kada igrač izgubi otvara se izbornik u kojem piše broj prikupljenih bodova te pomoću kojeg se korisnik može ponovo okušati u prelaženju igre pritiskom tipke „**Restart**“. Korisnik se može vratiti na početni izbornik, kao što je prije navedeno, pomicanjem miša prema lijevom rubu ekrana i pritiskom na tipku „**Back**“.



Slika 2.5 Prikaz izbornika u slučaju kada igrač izgubi

## 2.2 Neuronske mreže

Kao što je napisano u opisu proizvoda, kao reprezentaciju računalnog igrača smo koristili unaprijednu neuronsku mrežu (engl. Feed forward neural network).

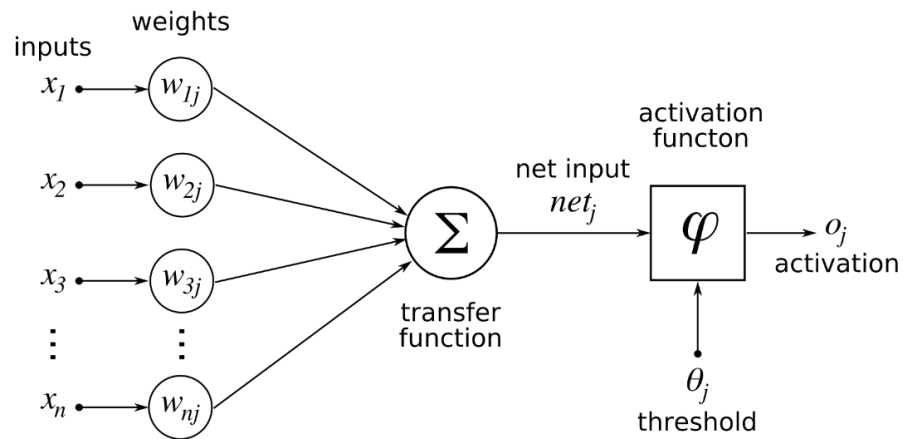
### 2.2.1 Potpuno povezana mreža

Kao genotip za potpuno povezanu neuronsku mrežu smo koristili polje realnih brojeva koji su predstavljali težine između dva neurona neuronske mreže.

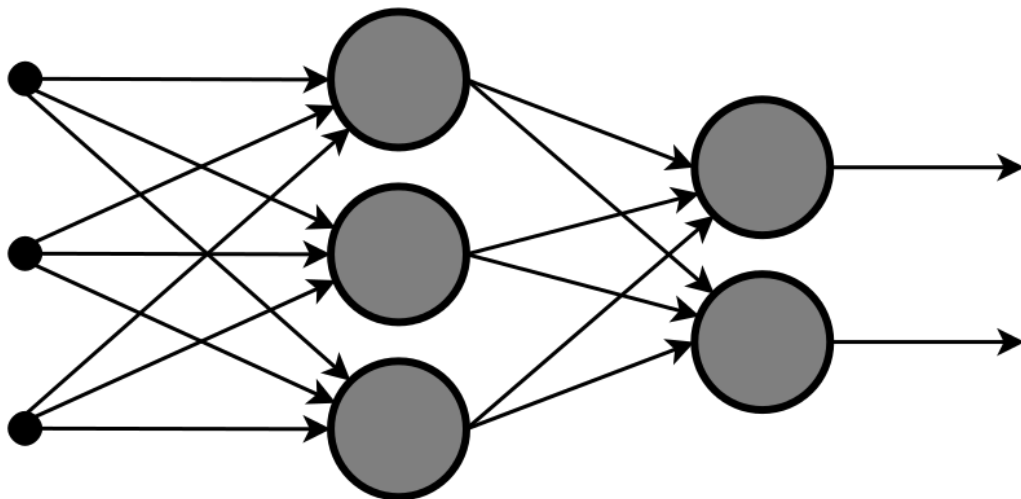
Neuroni u neuronskoj mreži su kao aktivacijske funkcije koristili sigmoidalnu funkciju, a kao transfer funkciju su koristili sumu svih težina.

$$f(x) = \frac{1}{1+e^{-net}}, net = \sum_{i=0}^d \omega_i x_i \quad (1)$$

|  |                  |
|--|------------------|
| Učenje neuronske mreže evolucijskim algoritmima na primjeru igre | Verzija: 1.0     |
| Neslužbena dokumentacija   | Datum: 24.1.2016 |



Slika 2.6 Model neurona neuronske mreže



Slika 2.7 Potpuno povezana unaprijedna neuronska mreža

Zbog jednostavne podatkovne strukture neuronske mreže mogli smo vrlo lako koristiti više različitih implementacija operatora križanja i mutacije poput:



|  |                  |
|--|------------------|
| Učenje neuronske mreže evolucijskim algoritmima na primjeru igre | Verzija: 1.0     |
| Neslužbena dokumentacija   | Datum: 24.1.2016 |

| Križanje                   | Mutacija   |
|----------------------------|--|
| Uniformno križanje         | Promjena vrijednosti gena po Gauss-ovoj distribuciji                       |
| Slučajan odabir gena       | Promjena vrijednosti gena po Gauss-ovoj distribuciji s nekom vjerojatnošću |
| BLX- <i>alpha</i> križanje | Promjena vrijednosti gena za neku konstantnu vrijednost                    |

Tablica 2-1 Tablica operatora križanja i mutacije

Najbolje rezultate smo dobili korištenjem BLX-*alpha* križanja u kombinaciji s mutacijom koja mijenja vrijednost nekog gena po Gauss-ovoj distribuciji, s definiranom vjerojatnošću.

|  |                  |
|--|------------------|
| Učenje neuronske mreže evolucijskim algoritmima na primjeru igre | Verzija: 1.0     |
| Neslužbena dokumentacija   | Datum: 24.1.2016 |

## 2.3 Evolucijski algoritmi

### 2.3.1 Klasični genetski algoritam

Za razvoj računalnog igrača su korišteni genetski algoritmi. Najbolji rezultat je dobiven korištenjem klasičnog genetskog algoritma koji je koristio sljedeću funkciju dobrote:

$$fit = broj\ prođenih\ cijevi + broj\ skupljenih\ novčića(nagrada) \quad (2)$$

Algoritam je koristio kao operator selekcije turnirsku selekciju.

```

1.  t ← 0
2.  InicijalizirajPopulaciju( P[t] );
3.  EvaluirajPopulaciju( P[t] );
4.  Dok ( nije zadovoljen uvjet zaustavljanja ) radi {
5.    P1 = OdaberiRoditelja( P[t] );
6.    P2 = OdaberiRoditelja( P[t] );
7.    C = KrižajRoditelje( P1, P2 );
8.    Mutiraj( C );
9.    DodajUPopulaciju( P', C );
10. EvaluirajPopulaciju( P[t'] );
11. t = t + 1;
12. P[t] = P';
13. } Završi

```

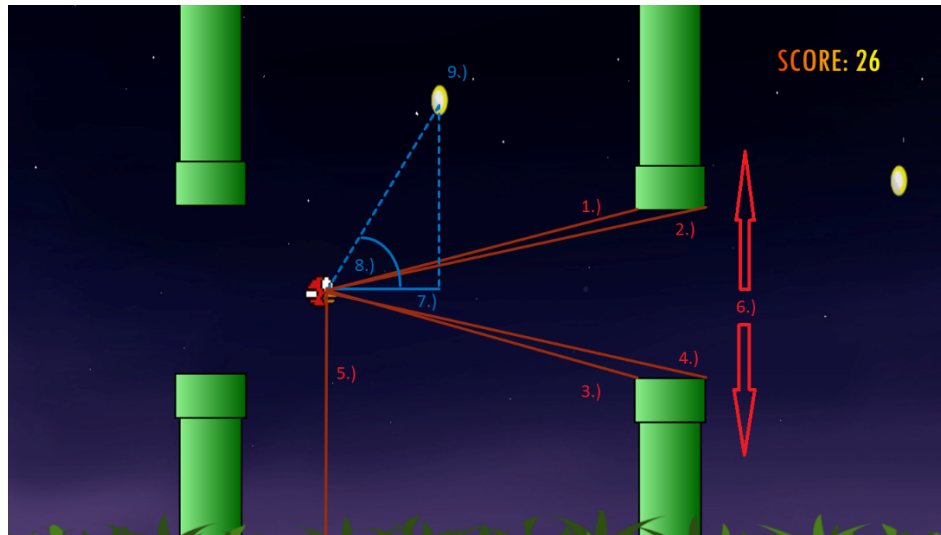
Algoritam 2-1 Klasični genetski algoritam

## 2.4 Ulazi u neuronsku mrežu

Igrač (Ptica) dobije neku reprezentaciju svijeta oko sebe na temelju koje treba zaključiti treba li skočiti ili ne. Tu informaciju je igrač dobivao svaki vremenski okvir (eng. *frame*). Isprobali smo različite implementacije igrača, a najbolje rezultate su ostvarili igrači koji su dobili sljedeće informacije:

1. Udaljenost igrača od bližeg kuta gornje zapreke
2. Udaljenost igrača od daljnjeg kuta gornje zapreke
3. Udaljenost igrača od bližeg kuta donje zapreke
4. Udaljenost igrača od daljnjeg kuta donje zapreke
5. Udaljenost igrača od dna, odnosno visina na kojoj se nalazi igrač
6. Smjer gibanja prve sljedeće zapreke
7. Udaljenost igrača po horizontalnoj osi od nagrade
8. Kut između horizontalne osi i pravca koji prolazi kroz centar ptice i centar nagrade
9. Zastavica koja govori postoji li u trenutnom odsječku igre nagrada

|  |                  |
|--|------------------|
| Učenje neuronske mreže evolucijskim algoritmima na primjeru igre | Verzija: 1.0     |
| Neslužbena dokumentacija   | Datum: 24.1.2016 |



Slika 2.8 Prikaz ulaza u neuronsku mrežu

#### Kako smo dosli do bas tih ulaza?

U početnim implementacijama nismo imali 9 ulaza već smo kroz razvoj projekta mijenjali i dodavali ulaze, u početku smo imali 4 ulaza(navedene stavke; prva, treća, peta i deveta). Kasnije smo dodali i nagrade u obliku novčića koje igrač može skupljati za dodatne bodove. Uz to smo dodali još dva ulazna parametra: udaljenost nagrade od igrača te zastavica koja pokazuje nalazi li se nagrada u trenutnom odsječku igre u kojem se nalazi igrač (gdje se odsječak odnosi na prostor između zadnje zapreke koje je igrač uspio izbjeći i zapreke koje još more izbjeći). Nakon toga smo dodali nove ulaze, koji su navedeni kao druga i četvrta stavka. Te konačno da bi pospjegli rezultate učenja neuronske mreže maknuli smo s ulaza udaljenost igrača od nagrade i dodali dva nova ulaza, a to su prethodno navedene sedma i osma stavka.

|  |                  |
|--|------------------|
| Učenje neuronske mreže evolucijskim algoritmima na primjeru igre | Verzija: 1.0     |
| Neslužbena dokumentacija   | Datum: 24.1.2016 |

### 3. Korištene tehnologije

#### 3.1 Programski jezik Java

Koristili smo programski jezik Java. Ovdje smo koristili novitet koji dolazi od Java 8 (1.8 službene) verzije, a to su lambda izrazi. Lambda izraze možemo koristiti prilikom stvaranja anonimnih razreda koji implementiraju sučelja koja su anotirana kao funkcijska sučelja, znači koja sadrže samo jednu metodu. Također koristili smo i još jedan bitan novitet koji dolazi s verzijom Java 8, a to je tokovni (engl. *streaming*) API kod kolekcija, pomoću kojeg se na učinkovit način može obavljati slijedna ili paralelna obrada podataka. Navedene novitete smo ekstenzivno koristili kod implementacije logike modela svijeta igre, čime smo postigli efikasnost i nadamo se veću čitljivost koda.

#### 3.2 Tehnologija JavaFX

JavaFX je Java tehnologija, namijenjena za izradu grafičkog korisničkog sučelja (engl. *GUI*). Omogućava programerima učinkovitu i brzu izradu bogatih platformski neovisnih aplikacija. Od svojih početaka, JavaFX koristeći prednost modernih grafičkih kartica kroz hardverski ubrzanu grafiku pruža skup dobro oblikovanih programskih sučelja koja omogućuju jednostavno kombiniranje grafike, animacije te naredba korisničkog sučelja.

Kod implementacije igre tehnologija JavaFX je korištena u izradi grafičkog korisničkog sučelja, grafičkog prikaza modela svijeta igre, ali i kod implementacije logike modela svijeta. Zbog bogate biblioteke koje sama tehnologija nudi odlučili smo komponente modela svijeta izvesti iz gotovih razreda JavaFX:

- `javafx.scene.shape.Rectangle` – korišten za prikaz zapreka
- `javafx.scene.shape.Circle` – korišten za prikaz igrača i nagrada

Zbog činjenice da se navedeni razredi mogu koristiti **neovisno o grafičkom prikazu**, upravo to nam je pogodovalo da takvu implementaciju modela svijeta iskoristimo za efikasno učenje neuronske mreže bez iscrtavanja komponenata na ekranu te kasnije za iscrtavanje i prikaz komponenata na ekranu tijekom izvođenja igre.

|  |                  |
|--|------------------|
| Učenje neuronske mreže evolucijskim algoritmima na primjeru igre | Verzija: 1.0     |
| Neslužbena dokumentacija   | Datum: 24.1.2016 |

## 4. Literatura

1. Čupić - Prirodom inspirirani optimizacijski algoritmi. Metaheuristike ( 2003 )
2. Čupić, Bašić, Šnajder - Umjetne neuronske mreže ( 2008 )
3. Čupić, Dalebo Bašić, Golub – Neizrazito, evolucijsko i neuroračunarstvo ( 2013 )
4. Millington – Game physics engine developement ( 2007 )
5. Jackson – Beginning Java 8 Games Development ( 2014 )

## 5. Tablica sadržaja

### 5.1 Slike

|  |    |
|--|----|
| Slika 2.1 Početni izbornik .....                             | 4  |
| Slika 2.2 Model svijeta dok igra korisnik .....              | 5  |
| Slika 2.3 Model svijeta dok igra neuronska mreža .....       | 5  |
| Slika 2.4 Izbornik postavki .....                            | 6  |
| Slika 2.5 Prikaz izbornika u slučaju kada igrač izgubi ..... | 7  |
| Slika 2.6 Model neurona neuronske mreže .....                | 8  |
| Slika 2.7 Potpuno povezana unaprijedna neuronska mreža ..... | 8  |
| Slika 2.8 Prikaz ulaza u neuronsku mrežu .....               | 11 |

### 5.2 Tablice

|   |   |
|---|---|
| Tablica 2-1 Tablica operatora križanja i mutacije ..... | 8 |
|---|---|

### 5.3 Algoritmi

|   |    |
|---|----|
| Algoritam 2-1 Klasični genetski algoritam ..... | 10 |
|---|----|