

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

**3. laboratorijska vježba iz Računalne grafike**

# **Primjena algoritma roja čestica na bojanje elemenata**

*Filip Kujundžić*

Zagreb, siječanj 2021.

# SADRŽAJ

<b>1. Uvod</b>	<b>1</b>
<b>2. Opis vježbe</b>	<b>2</b>
<b>3. Zaključak i sažetak</b>	<b>5</b>

# 1. Uvod

Računalne su igre proizvodi s najviše profita u industriji zabave. Njihov je razvoj s vremenom postao iznimno kompleksan, pa cijeli timovi rade na pojedinom naslovu. Dio tog procesa je i dizajn likova i predmeta koji se pojavljuju u igrama. Mnogi se algoritmi koriste kako bi se navedeni proces učinio efikasnijim. Jedan od njih je i algoritam roja čestica koji se koristi u alatu Massive za određivanje kretanja likova u masovnim scenama kao što su velike bitke (jedan od poznatih filmova u kojima je korišten je Avengers: Endgame). No, algoritam roja čestica može se upotrijebiti i u bojanju elemenata od kojih su napravljeni likovi. Upravo takav primjer bojanja lika sastavljenog od jednostavnih geometrijskih elemenata tema je ove laboratorijske vježbe.

## 2. Opis vježbe

Za demonstraciju primjene algoritma roja čestica na bojanje elemenata odabran je lik snjegovića koji se sastoji od trinaest elemenata. Ti su elementi: glava, tijelo, noge, šešir, lijevo oko, desno oko, nos, usta\_1, usta\_2, usta\_3, usta\_4, lijeva ruka i desna ruka. Usta\_i,  $i \in \{1, \dots, 4\}$  predstavljaju krugove koji su s lijeva na desno postavljeni da tvore usta snjegovića. Osnovna ideja je simulacija situacije u kojoj dizajner računalne igre treba odabrati izgled lika (u našem jednostavnom slučaju mora izabrati kombinaciju boja) a da se pri tome poštuju neka ograničenja.

Formulacija problema: Bojati snjegovića s bojama iz skupa deset različitih boja tako da se poštuju sljedeća ograničenja:

1. Broj upotrijebljenih boja mora biti veći ili jednak 4.
2. Boja glave i očiju ne smije biti jednaka.
3. Boja glave i nosa ne smije biti jednaka.
4. Boja usta mora biti različita od boje glave.
5. Boja šešira ne smije biti crna.
6. Glava ne smije biti narančaste ili koraljne boje.
7. Glava ne smije biti iste boje kao i šešir.

Posljednja su dva ograničenja dodana zbog estetskih razloga. Cilj je dobiti rješenje koje ispunjava što više zadanih ograničenja. Funkcija dobrote definirana je tako da za svako ispunjeno ograničenje vraća broj 1 a za neispunjeno broj 0. Dobivene se vrijednosti sumiraju te se traži njihov maksimum.

Pseudokod algoritma roja čestica izgleda ovako:

```
ulaz: populacija čestica:  $x_i$ 
izlaz: najbolje rješenje  $x_{best}$  i odgovarajuća vrijednost  $f_{min} = \min(f(x))$ 
1: inicijaliziraj čestice;
2:  $eval = 0$ ;
3: sve dok uvjet zaustavljanja nije ispunjen
4:   za  $i = 1$  do  $N_p$  ponavljaj
5:      $f_i = \text{evaluiraj\_novo\_rjesenje}(x_i)$ 
6:      $eval = eval + 1$ 
7:     ako  $f_i \leq pBest_i$  tada
8:        $p_i = x_i$ ;  $pBest_i = f_i$ ; // spremanje lokalno najboljeg rješenja
9:     ako  $f_i \leq f_{min}$  tada
10:       $x_{Best} = x_i$ ;  $f_{min} = f_i$ ; // spremanje globalno najboljeg rješenja
11:     $x_i = \text{generiraj\_novo\_rjesenje}(x_i)$ ;
```

Svaka čestica ima 13 dimenzija, od kojih svaka predstavlja boju kojom će jedan dio snjegovića biti bojan. Izgled čestice je sljedeći:

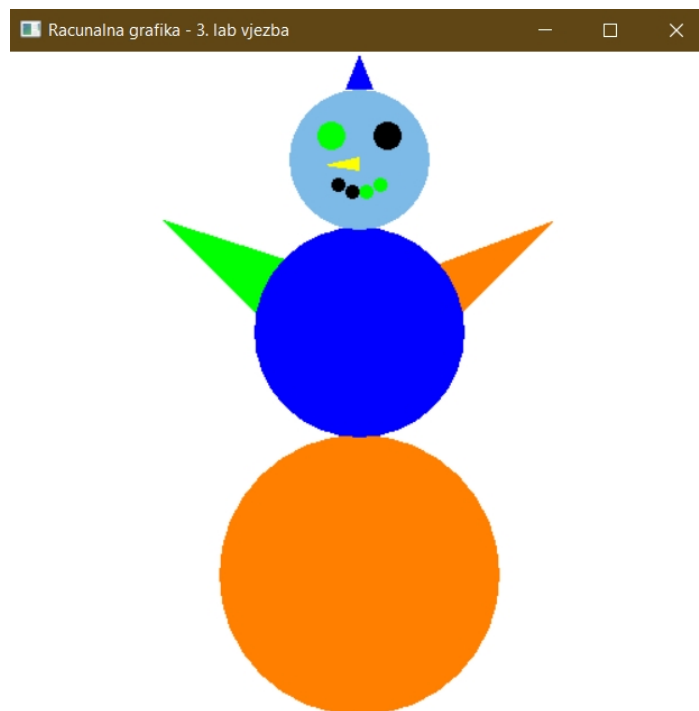
čestica = (glava, tijelo, noge, šešir, lijevo oko, desno oko, nos, usta\_1, usta\_2, usta\_3, usta\_4, lijeva ruka i desna ruka) Važno je napomenuti da je ovaj raspored dijelova snjegovića u pojedinim česticama statičan kako bi bilo lakše napisati ograničenja te ih potom mapirati na boje. Nakon što se napravi inicijalizacija čestica, za svaku česticu u populaciji radimo evaluaciju te ako je rješenje bolje od trenutnog najboljeg, spremamo ga ako najbolje lokalno, a ako je bolje od minimuma (u našem slučaju maksimuma) spremamo ga kao najbolje globalno rješenje. Čestica koja je najlošija se izbacuje iz populacije, a na njeno mjesto dolazi nova generirana čestica. Čestice pretražuju prostor veličine  $1 \times 1$  kako bi se dobivene vrijednosti mogle mapirati u boje. Novi položaj računa se kao suma starog i brzine koja između ostalog ovisi o društvenoj i spoznajnoj komponenti čestice. Nakon što je pronađena čestica koja ispunjava maksimalan mogući broj ograničenja, provodi se kodiranje u boje. U laboratorijskoj vježbi za algoritam roja čestica korišteno je 13 čestica, maksimalan dopušteni broj generacija nakon kojeg se prekida izvođenje algoritma u slučaju nepronaska rješenja je 100 te je za optimizaciju korištena maksimizacija.

Pokretanje programa:

Za izradu laboratorijske vježbe korišten je PyOpenGL i NiaPy framework za prirodom inspirirane optimizacijske algoritme te ih je potrebno instalirati prije pokretanja. Nakon toga, naredbom `python RG – LAB3.py` program se pokreće i dobivamo ispis sličan ovome:

```
7.0
[0.84010252 0.2546827 0.07693378 0.23442156 0.5400015 0.1254859 0.43904314
0.54831458 0.57674284 0.1807239 0.10636027 0.55996225 0.09884457]
Solution to color: [9, 3, 1, 3, 6, 2, 5, 6, 6, 2, 2, 6, 1]
```

Prvi broj, 7.0, nam govori koliko je ograničenja ispunjeno. Sva su ograničenja jednostavna, pa je algoritam uspješno pronašao barem jedno rješenje koje ih sve ispunjava. U uglatim zagradama vidimo vrijednosti čestice koja je odabrana kao najbolja, a u zadnjem retku imamo brojeve mapirane na boje kojima ćemo bojati snjegovića.



**Slika 2.1:** Snjegović bojan algoritmom

Jedno od mogućih rješenja programa prikazano je slikom iznad. Možemo vidjeti da je svih 7 ograničenja ispunjeno. Broj upotrebljenih boja veći je ili jednak 4. Donji dio je narančast, tijelo je plavo, lijeva ruka je zelena, desno oko je crno i nos je žut. Boja glave i očiju nije jednaka, glava je u boji "Maya Blue" a oči su zelene i crne boje. Glava i nos su također različitih boja, "Maya Blue" i žute, dok usta nisu "Maya Blue". Šešir je plavi, ali u drugoj nijansi nego glava. Boja glave nije niti narančasta niti koraljna.

### 3. Zaključak i sažetak

Izuzetno brzim razvojem računalnih igara povećava se potražnja za različitim načinima rješavanja problema s kojima se susrećemo prilikom razvoja istih. Prirodom inspirirani optimizacijski algoritmi pokazuju se kao jedno od učinkovitih rješenja u grafičkom dizajnu likova. U laboratorijskoj vježbi korišten je algoritam roja čestica u rješavanju problema s ograničenjima kojima smo odredili kako snjegović može biti bojan. Programsko rješenje je dalo izvrstan rezultat, no važno je napomenuti kako se radilo o jednostavnom zadatku i mekim ograničenjima. Povećanjem složenosti i broja ograničenja, programsko rješenje bi bilo nešto manje uspješno, ali još uvijek efikasno.

Literatura: Particle swarm optimization for automatic creation of complex graphic characters; Iztok Fister Jr., Matjaž Perc, Karin Ljubič, Salahuddin M.Kamal, Andres Iglesias, Iztok Fister