

Fakultet elektrotehnike i računarstva

Računalna grafika, 3. lab. Vježba:

WebGL istraživač Mandelbrotovog skupa
korištenjem sjenčara

Petar Mihalj, 0036497900

Prosinac 2020.

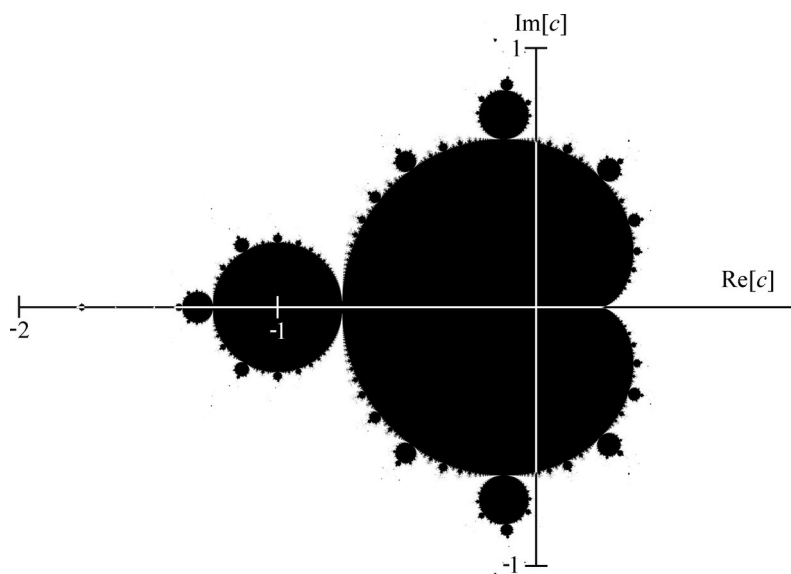
1. Matematička pozadina

Mandelbrotov skup definiramo kao skup kompleksnih brojeva, za koje iteracijska formula

$$z_n = z_{n-1}^2 + c$$

sa početnim uvjetom $z_0=0$ (c predstavlja broj kojem testiramo pripadnost u Mandelbrotovom skupu) ne divergira.

Budući da ova definicija nije pretjerano prilagođena za praktično testiranje pripadnosti, koristimo činjenicu da svi kompleksni brojevi kojima je apsolutna vrijednost veća od 2 nužno divergiraju. Naravno, ako u bilo kojem koraku iteracije z postane apsolutne vrijednosti veće od 2, iteracijska formula će divergirati.



Slika 1: Mandelbrotov skup
(<https://math.stackexchange.com/questions/1732960>)

Na Slici 1 vidljiv je prikaz Mandelbrotovog skupa u kompleksnoj ravnini. Crni dijelovi predstavljaju one točke koje nakon nekog fiksnog broja iteracija nisu divergirale u osnovnoj formuli.

Alternativni prikaz M. skupa temelji se na bojanju i onih točaka koje su divergirale, ali tako da boja ovisi o broj iteracija koje su točki trebale da izađe iz kruga radijusa 2, centriranog u nuli.

Jasno je da za svako izračunavanje izgleda M. skupa treba iterirati istu formulu na mnogo točaka; ovo je idealan problem za korištenje sjenčara, minimalnih programa koji se izvršavaju na grafičkoj kartici.

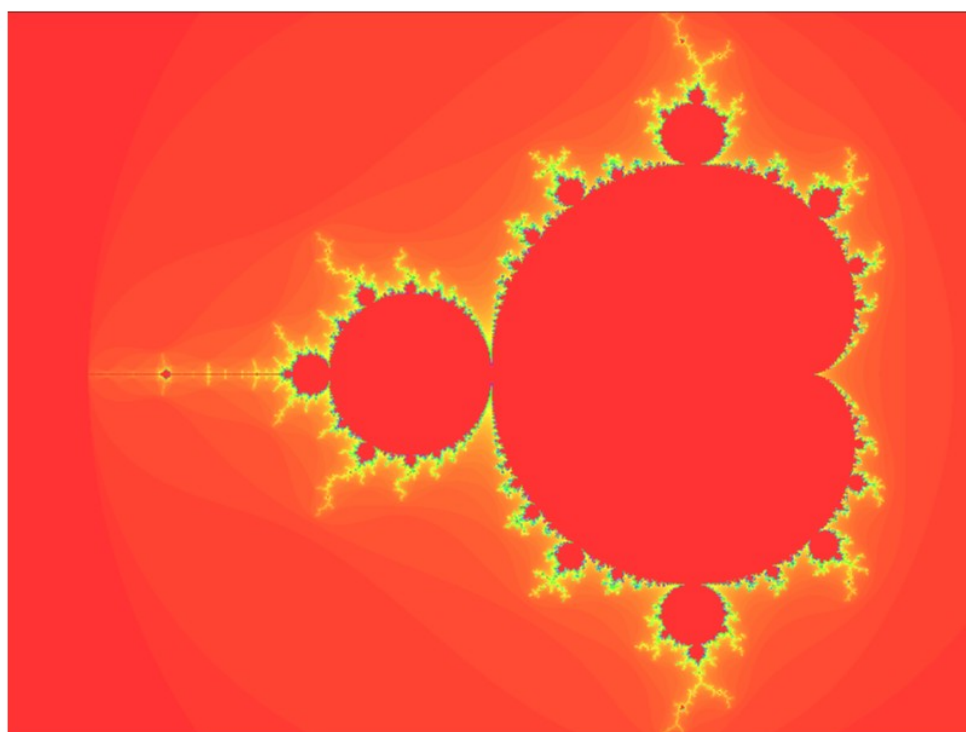
2. Funkcionalnosti sustava

WebGL istraživač M. skupa ostvaren je korištenjem WebGL-a, grafičke knjižnice namijenjene korištenju u web preglednicima.

Korisničko sučelje nudi intuitivne komande za zumiranje skupa.

Sustav je respozivan čak i sa relativno visokom rezolucijom (800x600), vrijeme za crtanje jedne scene ispod je pola sekunde, čak i kad je pozadinski procesor Intel i7, a ne dedikirana grafička kartica.

Mandelbrot Explorer, Petar Mihalj (2020)



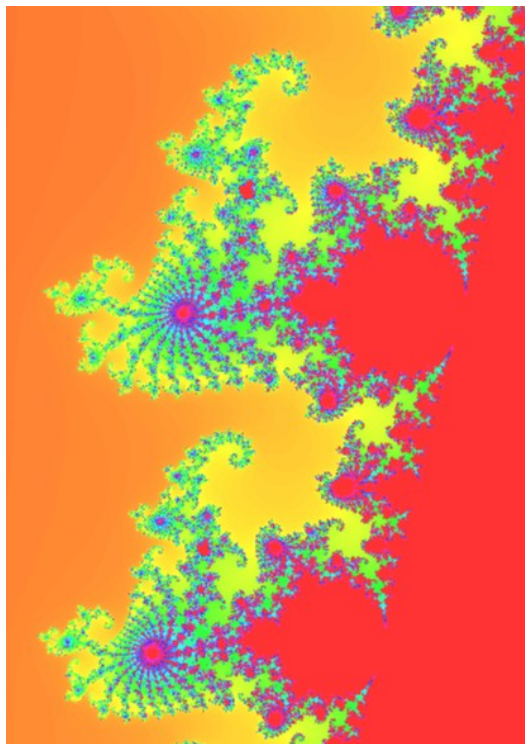
Click to zoom in

Press SPACE to return back.

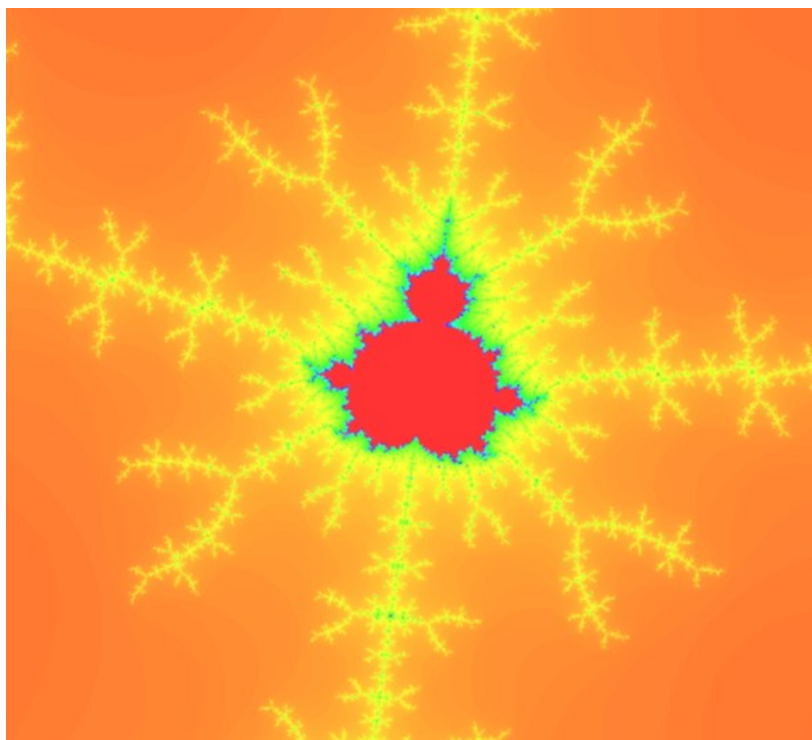
Press ESCAPE to restart.

Slika 2: Korisničko sučelje

U nastavku su dani neki zanimljivi dijelovi skupa:



Slika 3



Slika 4

Na slici 4 uočavamo zanimljiv fenomen; Mandelbrotov skup u sebi sadrži dijelove koji sliče njemu samom! Ovo je naravno razlog zašto ga svrstavamo u fraktale.

3. Analiza sjenčara

3.1. Sjenčar vrhova (vertex shader):

```
1 precision highp float;
2 attribute vec2 coords;
3 attribute vec2 c;
4
5 varying float count;
6
7 precision highp int;
8 uniform int max_iters;
9
10 void main(void) {
11     vec2 z = vec2(0.0,0.0);
12     vec2 z_new = vec2(0.0,0.0);
13
14     for(int i=0;i<1000000;++i){
15         if (i == max_iters){
16             count = float(max_iters);
17             break;
18         }
19         z_new[0] = z[0] * z[0] - z[1] * z[1] + c[0];
20         z_new[1] = z[0] * z[1] * 2.0 + c[1];
21         z = z_new;
22
23         float abs_val = z_new[0]*z_new[0]+z_new[1]*z_new[1];
24         if (abs_val >= 4.0){
25             count = float(i);
26             break;
27         }
28     }
29
30     gl_Position = vec4(coords, 0.0, 1.0);
31 }
```

Sjenčar vrhova zadužen je za provođenje iteracija nad svakom točkom. Broj iteracija zadan ograničen je sa čarobnom konstantom zbog ograničenja jezika GLSL (ograničena ekspresivnost for petlje). U svakoj iteraciji provjerava se je li točka izašla iz kruga radijusa 2. Ako je, petlja se prekida. Primijetite razliku između c i coords; c je kompleksni broj, a coords označava koordinate točke u $[-1, 1] \times [-1, 1]$ prostoru ekrana.

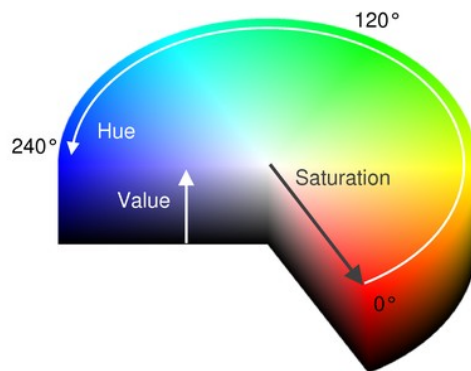
U sjenčar fragmenata implicitno šaljemo koordinate u prostoru ekrana, zajedno s brojem iteracija (varying float count).

3.1. Sjenčar fragmenata (fragment shader):

```
1 precision highp float;
2 precision highp int;
3 uniform int max_iters;
4 varying float count;
5
6 // https://github.com/hughsk/glsl-hsv2rgb
7 vec3 hsv2rgb(vec3 c) {
8     vec4 K = vec4(1.0, 2.0 / 3.0, 1.0 / 3.0, 3.0);
9     vec3 p = abs(fract(c.xxx + K.xyz) * 6.0 - K.www);
10    return c.z * mix(K.xxx, clamp(p - K.xxx, 0.0, 1.0), c.y);
11 }
12
13 void main(void) {
14     float max_iters_fl = float(max_iters);
15     float hue = count / max_iters_fl;
16     gl_FragColor = vec4(hsv2rgb(vec3(hue, 0.8, 1.0)), 1.0);
17 }
```

Sjenčar fragmenata obavlja relativno jednostavnu zadaću; broj iteracija linearno normalizira na interval $[0,1]$, a zatim taj interval preslikava u spektar boja.

Preslikavanje se temelji na prelasku u HSV reprezentaciju boja:



Slika 5: HSV (<https://dsp.stackexchange.com/a/30263>)

Možemo linearno interpolirati po našem subjektivnom doživljaju različitosti i sličnosti boja, i to tako da iteriramo po hue komponenti HSV-a. Naravno, takvim pristupom će se boje na kraju i početku intervala svesti na crvenu (hue = 0).

4. Upute za pokretanje:

Kod cijelog projekta, uključujući ovu dokumentaciju, dostupan je na:

<https://github.com/PetarMihalj/MandelbrotExplorer>

Nakon preuzimanje programskog koda, pokretajte ga otvaranjem main.html datoteke u svom web pregledniku. Naravno, preglednik mora imati podršku za WebGL.