

Vizualizacija Julijevega skupa

Laboratorijska vježba iz računalne grafike

Antun Magdić

22. siječnja 2021.

1 Teorijska pozadina

Matematički korektna definicija Julijevega skupa može se pronaći u [1]. U ovoj laboratorijskoj vježbi korišteni su Julijevi skupovi dobiveni iteracijom kvadratnih kompleksnih polinoma. Radi se o funkciji $f : \mathbb{C} \rightarrow \mathbb{C}$ oblika

$$f(z) = z^2 + c,$$

gdje je c neki kompleksan broj. Definiramo pravilo za generiranje niza (z_n) sljedećim izrazom

$$z_n = f(z_{n-1}) = z_{n-1}^2 + c.$$

Ovisno o prvom elementu niza z_0 ovaj niz može biti ograničen i u tom slučaju može konvergirati ili ne mora. Ako niz nije ograničen onda divergira. Definiramo skup $S(c)$ kao skup svih kompleksnih brojeva z_0 za koje niz (z_n) divergira. Julijev skup $J(c)$ tada je granica skupa $S(c)$. Julijev skup može biti povezan ili potpuno nepovezan, a to ovisi o divergenciji niza za $z_0 = 0$. Odavde dolazi veza Julijevega i Mandelbrotovog skupa [2], odnosno Julijev skup $J(c)$ je povezan, ako i samo ako c pripada Mandelbrotovom skupu.

Julijev skup jedan je od ključnih pojmova u teoriji dinamičkih sustava koji često pokazuju kaotično ponašanje. Osim u zbog svoje primjene u matematici Julijev skup postao je poznat i po svojoj estetskoj ljepoti i fraktalnoj strukturi.

2 Opis razvijenog programa

Razvijeni program crta i animira Julijev skup. Program se pokreće iz naredbenog retka sa dva argumenta: udaljenosti r kompleksnog broja c od ishodišta i opcijom `bw` ili `colorful`. Nakon pokretanja animacije počinje. Za početnu vrijednost parametra c uzima se r . Kako prolazi vrijeme c kruži oko ishodišta nekom frekvencijom ω , odnosno nakon što je prošlo t vremena od pokretanja animacije, vrijednost broja c je $re^{i\omega t}$. Na taj način c se stalno nalazi na udaljenosti r od ishodišta. Izgled Julijevega skupa se mijenja kako c kruži oko ishodišta i ovisno o tome pripada li c Mandelbrotovom skupu, Julijev skup je povezan ili nepovezan.

Za vrijeme rada programa, tipkama na tipkovnici moguće je upravljati slikom na ekranu. Dostupne su sljedeće mogućnosti:

- `<RAZMAKNICA>` pokreće i pauzira animaciju,
- `<STRELICA UDESNO>` premotava animaciju unaprijed,
- `<STRELICA ULJEVO>` premotava animaciju unatrag,
- `d` pomiče kameru u smjeru 1,
- `w` pomiče kameru u smjeru $+i$,
- `a` pomiče kameru u smjeru -1 ,

- **s** pomiče kameru u smjeru $-i$,
- **i** povećava sliku,
- **o** smanjuje sliku,
- **p** ispisuje trenutnu vrijednost c .

Na slici 1 prikazani su neki primjeri Julijevih skupova.

3 Implementacijski detalji

Program je implementiran koristeći tehnologiju *OpenGL* koristeći programski jezik *Julia*. Da bi se iscrtala slika za svaki slikovni element određuje se z_0 . Zatim se procjenjuje divergira li niz za određeni z_0 i na temelju toga se boja odgovarajući slikovni element. Za ovo je pogodno koristiti sjenčare budući da je divergencija niza za pojedini slikovni element potpuno nezavisna od ostalih slikovnih elemenata, pa je posao lako paralelizirati. Zbog toga je iscrtavanje vrlo brzo, i animacija se bez poteškoća odvija u stvarnom vremenu.

Da bi se procijenilo divergira li niz za neki z_0 , iterativno se određuju njegovi članovi. Ako za neki z_n vrijedi $|z_n| > R$, gdje je

$$R = \frac{1 + \sqrt{1 + 4|c|}}{2},$$

tada znamo da niz divergira i nema smisla raditi daljnje iteracije. U nekom trenutku treba stati s iteracijama i tada niz proglašavamo ograničenim (iako on to ne mora biti jer je moguće da tek za veliki n vrijedi $|z_n| > R$). Boja slikovnog elementa određuje se na temelju broja iteracija potrebnih da se pokaže divergencija. Moguće su razne strategije dodjele boje, ali u glavnom se radi o načelu pokušaja i pogrešaka, a krajnja strategija uvelike ovisi o ukusu autora.

Jedna od mogućih strategija je definiranje pravokutnika gdje se svakom vrhu dodjeljuje neka proizvoljna boja. Može se definirati funkcija $c : \mathbb{N} \rightarrow \mathbb{R}^2$ koja određenom broju iteracija pridružuje par realnih brojeva

$$(x, y) = c(i).$$

Na temelju dobivenih x i y i odabranih točaka pravokutnika bilinearnom interpolacijom određuje se konačna boja. Na slici 1 korištena je funkcija

$$c(i) = e^{-\gamma_c i} (\cos(\omega_c i), \sin(\omega_c i)),$$

gdje su γ_c i ω_c neki parametri, a točke pravokutnika su $(1, 1)$, $(-1, 1)$, $(-1, -1)$ i $(1, -1)$.

4 Upute za pokretanje

Program je testiran na računalu s operativnim sustavom *macOS* 10.15.7 i i programskim jezikom *Julia* verzije 1.5.3. Za pokretanje su potrebni moduli jezika *Julia CSyntax*, *GLFW* i *ModernGL*.

Nakon pozicioniranja u direktorij s izvornim kodom `juliaset.jl`, program iz naredbenog retka s dva argumenta:

```
julia juliaset.jl <r> <način rada>
```

gdje je `<r> = |c|`, a način rada može biti `colorful` ili `bw`. U direktoriju se moraju nalaziti i datoteke `fragment_shader_bw.glsl`, `fragment_shader_colorful.glsl` i `vertex_shader.glsl`.

References

- [1] Julia set, https://en.wikipedia.org/wiki/Julia_set
- [2] Mandelbrot set, https://en.wikipedia.org/wiki/Mandelbrot_set

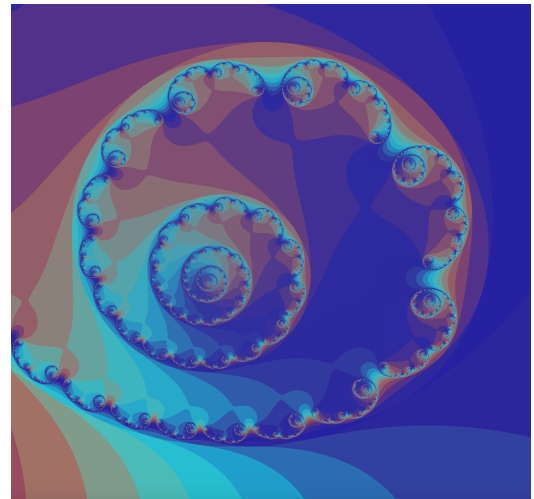
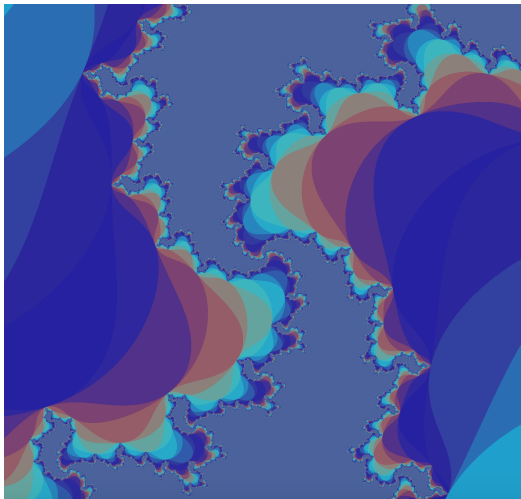
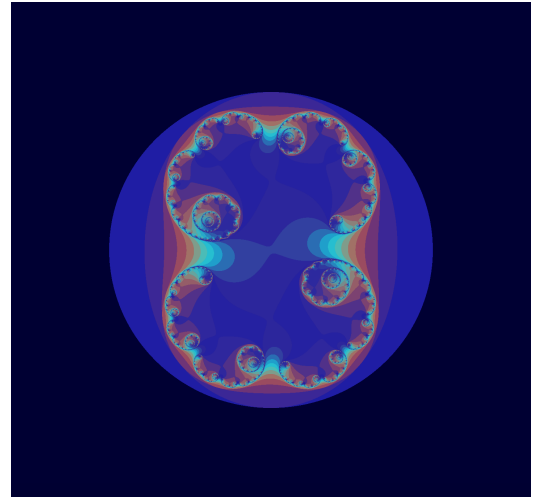
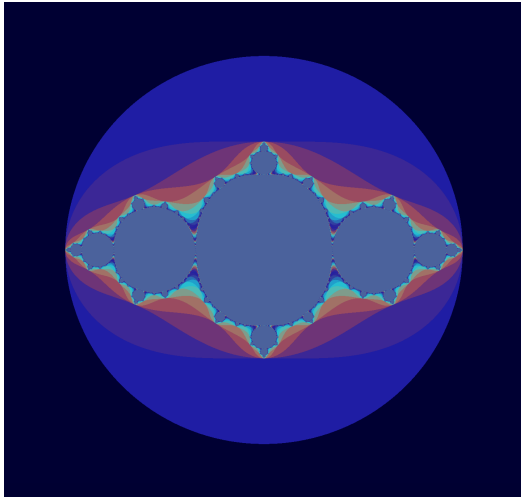
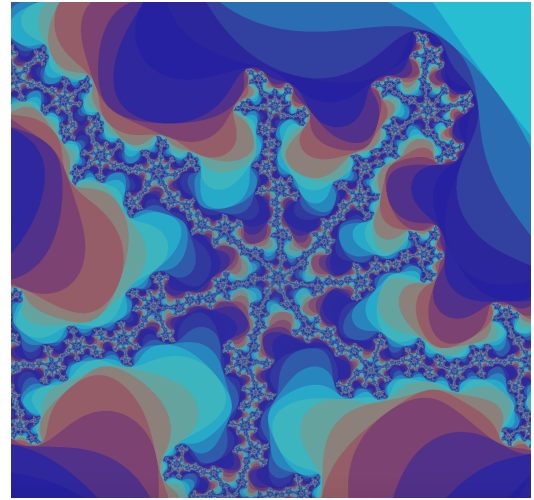
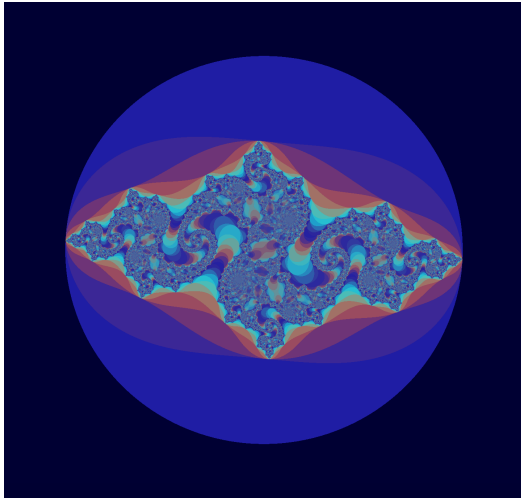


Figure 1: Neki primjeri iscrtanih Julijeveih skupova u načinu rada *colorful*.