

Kreativno kodiranje v generativni vizualizaciji

Davorin Babič, profesor medijske tehnologije

Mreža znanja 2018, 21. in 22. november



Kreativno kodiranje

- Strokovni modul v **odprtem kurikulumu** za transverzalne predmetne vsebine, ki se pojavljajo v določenih ravneh **medijske produkcije**:
 - digitalne kompetence,
 - reševanje problemov in izzivov,
 - kritično in ustvarjalno razmišljanje,
 - matematične kompetence,
 - sposobnost načrtovanja za doseganje zastavljenih ciljev,
 - timsko delo in sodelovalno učenje,

Kreativno kodiranje

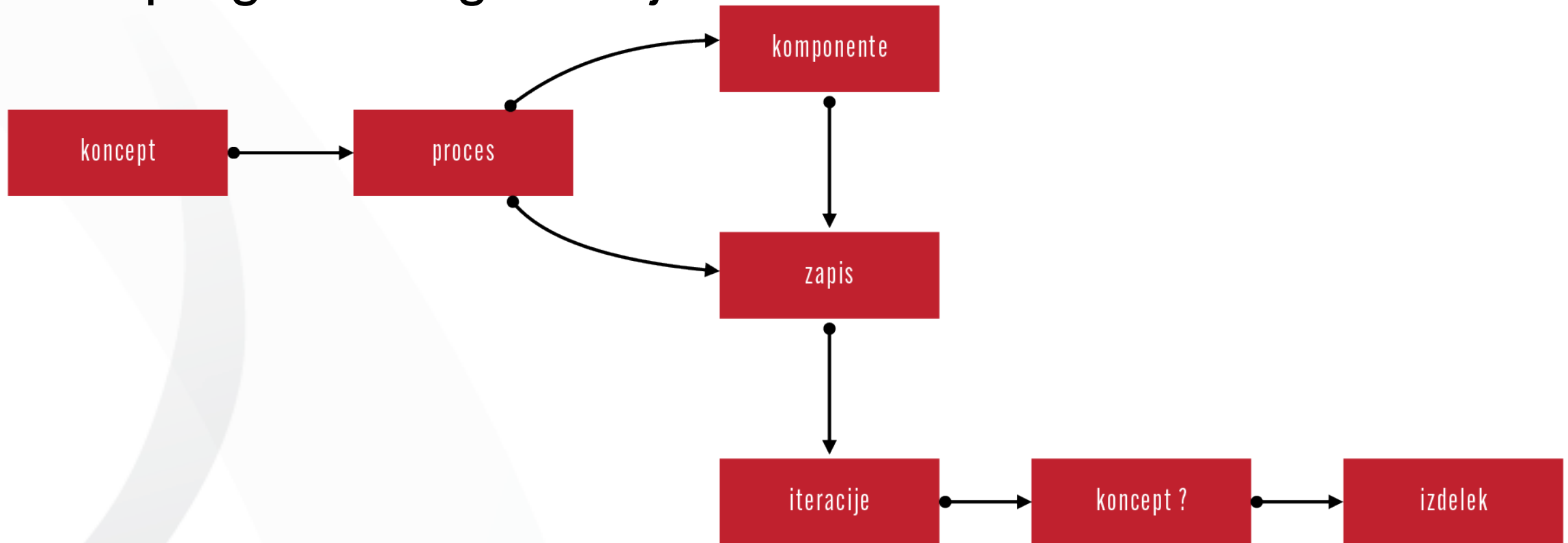
- celostni in izkustveni učni pristop v počevanju programiranja, v katerem povežemo osnovne matematične koncepte in strukture s koncepti sodobnega likovno–ustvarjalnega procesa,
- sposobnost ustvarjalnega izražanja zamisli v sodelovanju z ustreznim programskim orodjem kot eksplicitni del učnega okolja.

Generativna vizualizacija

- Predstavlja področje v sodobni umetnosti, ki se razvija skozi čas s pomočjo **avtonomnega sistema**, **računalniške kode**,
- dokaz, da **informacijska tehnologija** prodira tudi v umetniško ustvarjanje,
- **algoritemski** način ustvarjanja estetike,
- nov, zabaven pristop h **kodiranju**.

Generativni pristop

- Vizualna abstrakcija skozi generativni sistem programskega orodja



Pedagoški pristop

- Učitelj v vlogi **mentorja** za ustvarjanje učinkovitega učnega okolja kot motivacijski pristop izbere poučevanje po načelih **kombiniranega učenja** (angl. blended learning),
- dijakom omogoča različne vrste **aktivenga učenja**, zlasti sodelovalno učenje z metodo **kodiranja v paru**, **problemsko učenje** in **učenje s pomočjo ikt**,

Pedagoški pristop

- ustvarjajo se **skupna mnenja, znanje v paru,** skupinska (**parna**) **inteligenca,**
- poveča se **samozavest** in **motivacija** dijakov za delo,
- uvid smisla v učenju in uporabnosti znanj na osnovi **povezovanja različnih vsebin** pouka s poudarkom na **izkušenskem učenju.**

Kreativnost v kodiranju

- Sposobnost **kreativnega izražanja** zamisli in idej,
- kodiranje temelji na **konceptualnem znanju** večplastnosti digitalne pismenosti
 - medijska, tehnološka, vizualna,
 - **prenos** poznavanja prvin likovne teorije,
 - s pomočjo kodiranja reševanje likovnih **izzivov/problemov** z likovnim izražanjem,

Kreativnost v kodiranju

- **algoritmčno razmišljanje** kot medij za samoizražanje,
- razvoj **miselnih sposobnosti** in **metakognitivnih strategij** (kot del omrežja idej), ki omogočajo lažje reševanje omenjenih izzivov/problemov.

Miselne sposobnosti

- **Analiza + dekompozicija** izziva/problema na množico elementarnih, bolj obvladljivih korakov,
- **prepoznavanje, posploševanje in abstrakcija vzorcev**, ki se pri tem pojavljajo,

Miselne sposobnosti

- **sinteza** vseh prejšnjih komponent v delujoč in učinkovit **algoritem**,
- razumevanje hierarhije
 - izzivi/problemi/situacije,
 - koncepti,
 - algoritmi,
 - programska koda.

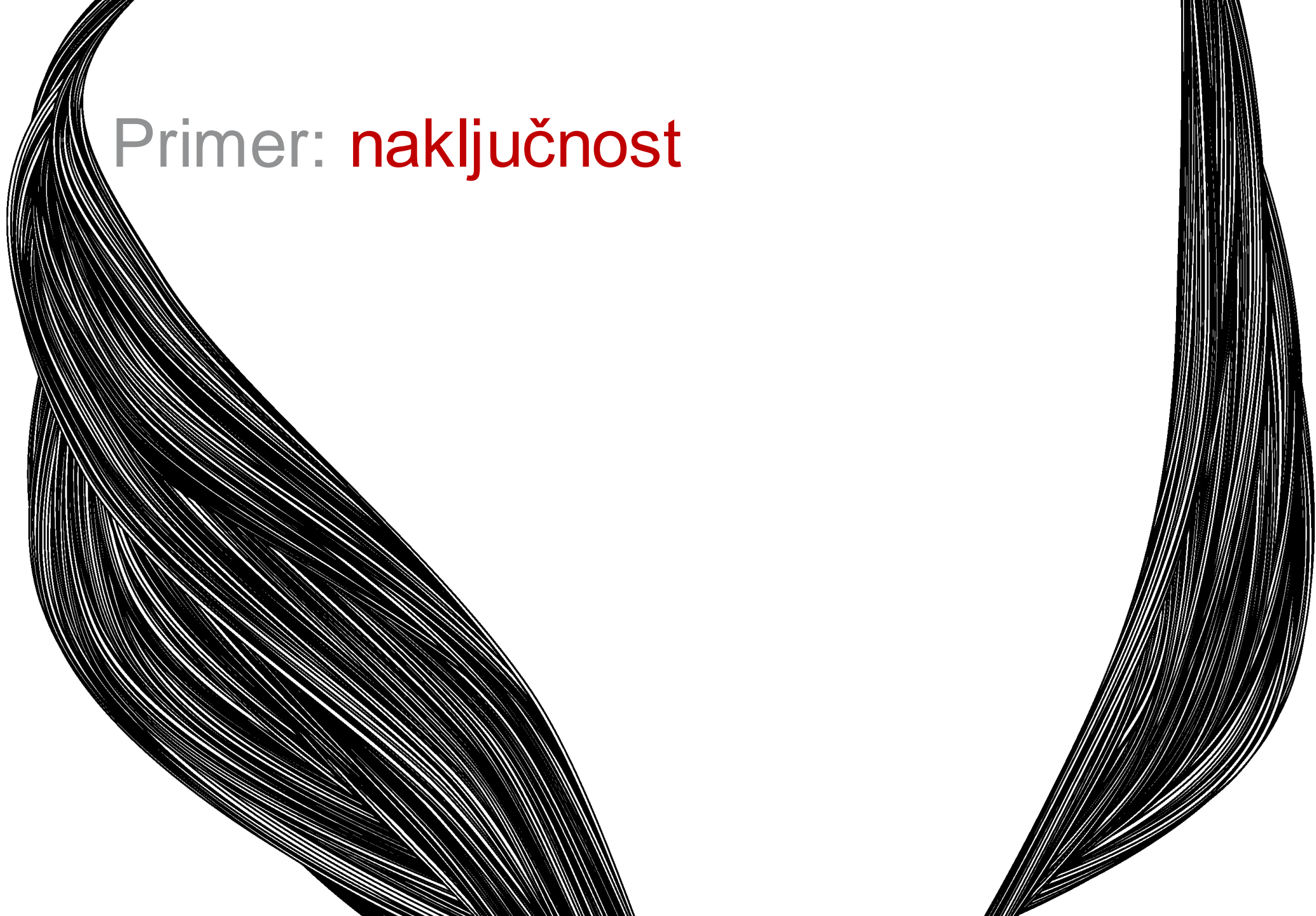
Matematični koncepti

- spoznavanje konceptov **pred** orodji, pred kodiranjem,
- kodiranje naredi matematiko **prijetnejšo**,
- matematika je **uporabna** v ustvarjalnih vajah in mini projektih,
- kodiranje kot medij, **računanje** kot ustvarjalni proces,
- **ustvarjalno partnerstvo** – oblikovalec + matematika + algoritem.

Processing

- OFRO, poenostavljeni programski jezik java in po meri dodane lastnosti, ki so povezane z **grafiko** in **interaktivnostjo**,
- razvit za **programsko skicirko** in poučevanje osnov računalniškega programiranja znotraj **vizualnega konteksta**,
- **učenje kodiranja** skozi **ustvarjanje grafike** – sledi takojšnja povratna informacija o ustvarjeni grafiki, spodbuja nadaljnje učenje.

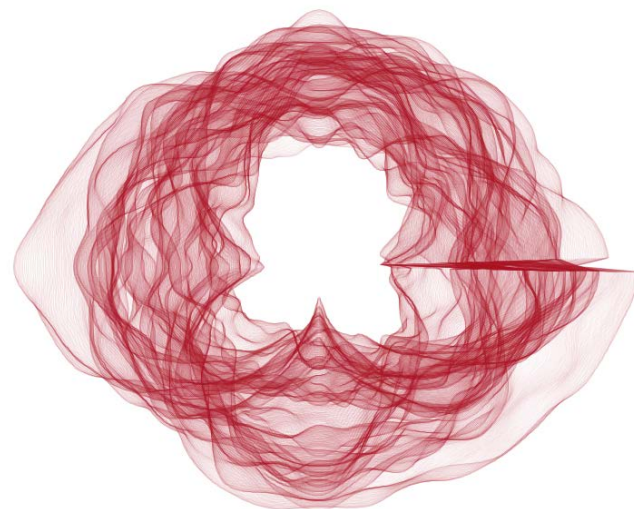
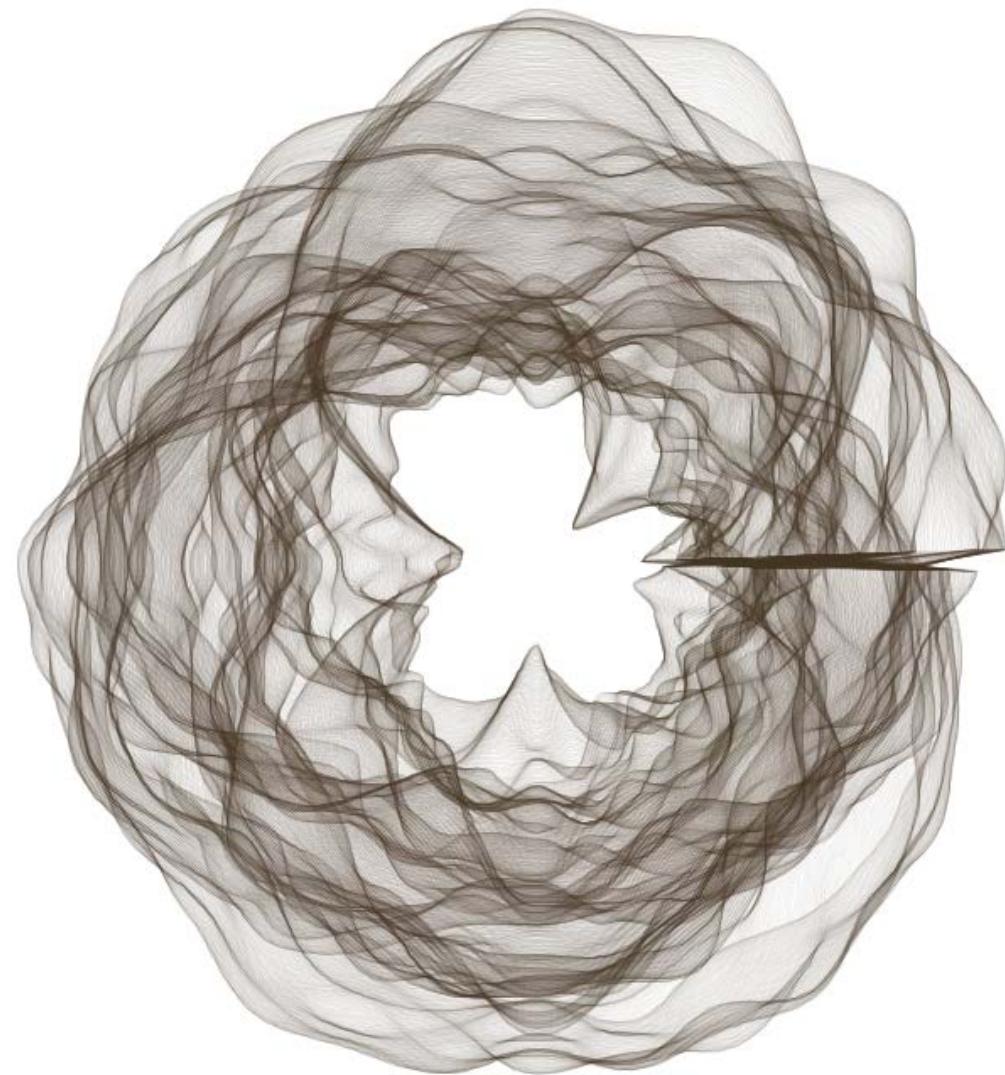
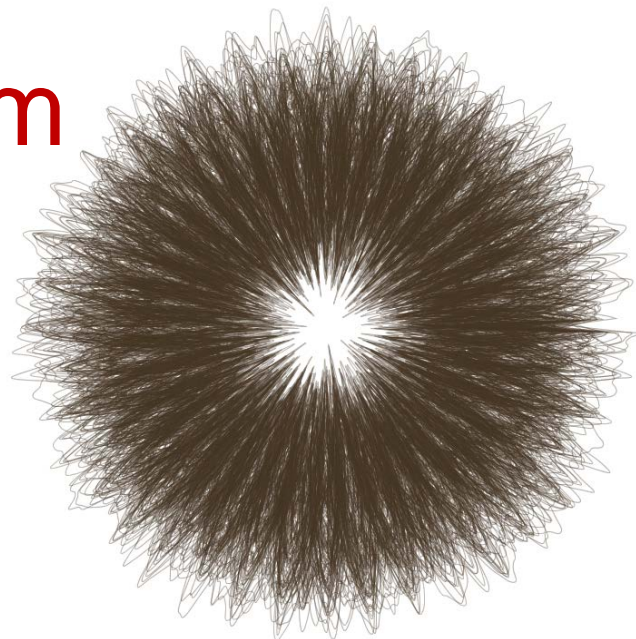
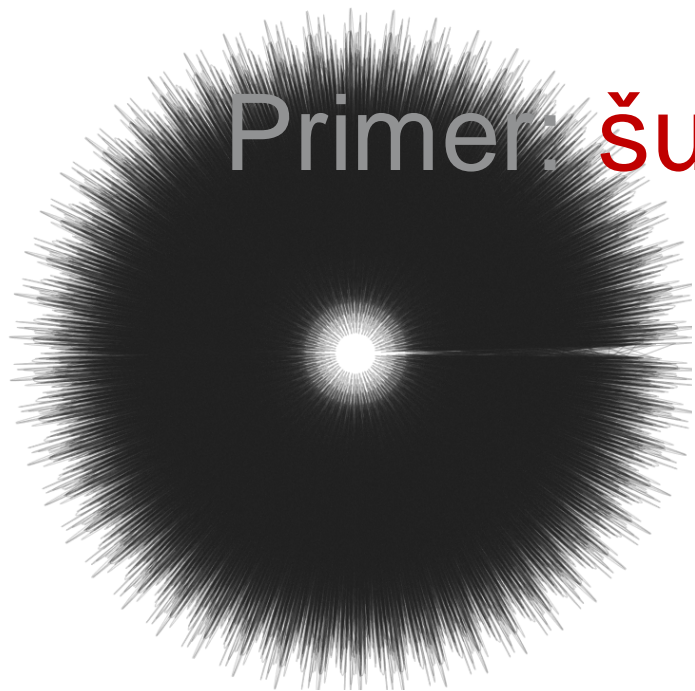
Primer: **naključnost**



```
void draw() {
    background(255);
    for (int i=0; i< iteracije; i++) {
        for (int a=0; a < stranice; a++) {
            x[a] += random(-varianca, varianca);
            y[a] += random(-varianca, varianca);
        }
        beginShape();
        curveVertex(x[stranice-1]+width/2, y[stranice-1]+height/2);
        for (int a=0; a < stranice; a++) {
            curveVertex(x[a]+width/2, y[a]+height/2);
        }
        curveVertex(x[0]+width/2, y[0]+height/2);
        curveVertex(x[1]+width/2, y[1]+height/2);
        endShape();
    }
    endRecord();
}
```



Primer: šum



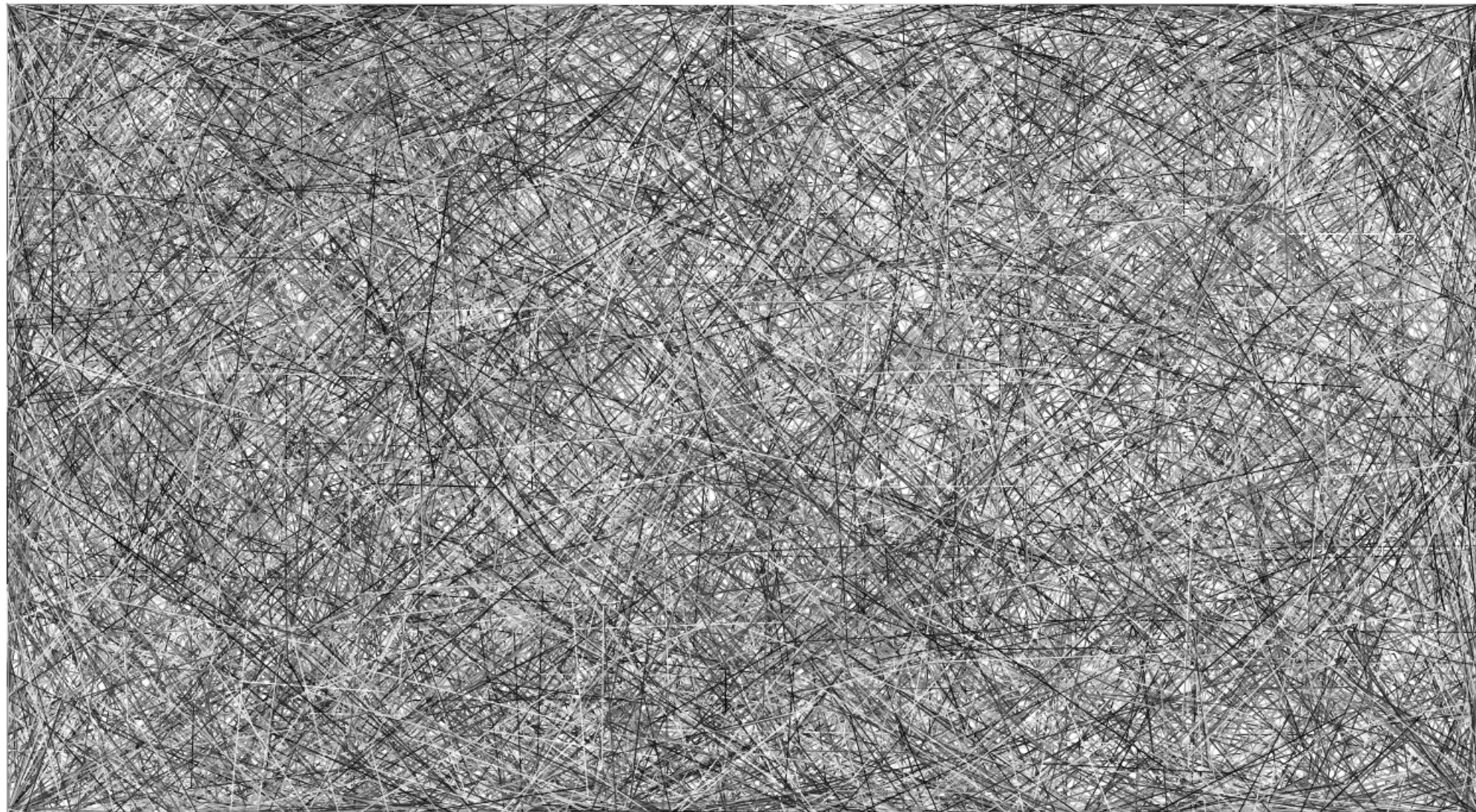

```
void draw() {
  translate(width/2, height/2);
  beginShape();
  for (int i = 0; i < 400; i++) {
    float kot = map(i, 0, 400, 0, TWO_PI);
    float r = 400 * noise(i * 2, t * 1);
    float x = r * cos(kot);
    float y = r * sin(kot);
    curveVertex(x, y);
  }
  endShape(CLOSE);

  t += 1;

  if (frameCount % 2000 == 0) {
    noLoop();
  }
}
```



Primer: random walk



```
void randWalk() {
    xk = random(-d, d);
    yk = random(-d, d);
    x = constrain(xc + xk, p, width - p);
    y = constrain(yc + yk, p, height - p);
    float r = random(255);
    stroke(r);
    line(xc, yc, x, y);
    xc = x;
    yc = y;
    if (frameCount % 60 == 0) {
        String s = "imgs/rW" + nf(stevilo, 5) + ".jpg";
        save(s);
        stevilo++;
    }
}
```



Spletarna

- www.processing.org
- <http://wiki.c2.com/?DesignByNumbers>
- <https://processingfoundation.org>
- <https://algenart.wordpress.com>