FAKULTA APLIKOVANÝCH VĚD
ZÁPADOČESKÉ UNIVERZITY
V PLZNI

Semestrální práce z KIV/PRO

Porovnání algoritmů Value Noise, Perlin Noise a Worley Noise

Autor: Vít Novotný

Datum: 28. listopadu 2025

Obsah

1 Zadání

Cílem práce je implementace algoritmů pro procedurální generování dvourozměrného šumu a jejich následné porovnání. Hlavní metodou je algoritmus **Perlin noise**, ke kterému jsou pro účely porovnání implementovány algoritmy **Value noise** a **Worley (cellular) noise**.

Všechny algoritmy budou implementovány v jazyce Python a porovnávány při stejných parametrech (seed, měřítko, rozlišení). Výstupem budou grayscale obrázky, analýza vlastností jednotlivých noise metod a jejich vzájemné porovnání.

2 Existující metody

2.1 Value Noise

Value noise je jednoduchý procedurální šum, ve kterém je každému rohu mřížky přiřazena pseudonáhodná skalární hodnota. Hodnota šumu uvnitř buňky se pak získává interpolací těchto čtyř hodnot, obvykle pomocí lineární nebo vyhlazené fade funkce. Výsledkem je hladký, vizuálně jednoduchý šum, který zachovává základní mřížkovou strukturu.

Výhodou value noise je jeho rychlá a snadná implementace a nízká výpočetní náročnost. Nevýhodou je omezená vizuální komplexnost a to, že výsledný šum může působit uměle nebo příliš pravidelně. Value noise se používá spíše jako úvodní metoda nebo základ pro složitější algoritmy, které z něj vycházejí. [2]

2.2 Perlin Noise

Perlin noise, který navrhl Ken Perlin v roce 1985, je gradientový šum. Místo náhodných hodnot přiřazuje každému rohu buňky náhodný gradientový vektor. Hodnota šumu se vypočítá jako skalární součin tohoto vektoru a směru k vyhodnocovanému bodu. Výsledný šum je následně plynule interpolován pomocí speciální vyhlazovací funkce.

Hlavní výhodou Perlinova šumu je jeho přirozený a spojitý vzhled, díky kterému se používá pro generování terénů, oblaků, ohně nebo organických textur. Nevýhodou je složitější implementace a možnost vzniku jemných mřížkových artefaktů. I přesto je Perlin noise dlouhodobě standardem v počítačové grafice a základní metodou mnoha dalších šumových algoritmů. [1]

2.3 Worley Noise

Worley noise, představený Stevenem Worleym v roce 1996, je šum založený na vzdálenosti. Každá buňka mřížky obsahuje náhodně umístěný feature point a hodnota šumu je určena vzdáleností k nejbližšímu z těchto bodů. Výsledné vzory připomínají Voroného diagramy nebo buněčné struktury.

Výhodou Worley noise je charakteristický, výrazně odlišný vzhled, vhodný pro textury jako kámen, pěna, popraskané povrchy nebo organické materiály. Nevýhodou je vyšší výpočetní náročnost a méně spojitý charakter výsledného šumu. Oproti Perlinovu šumu vytváří ostřejší a výraznější obrazce. [3]

3 Zvolené řešení

3.1 Sjednocené rozhraní

Pro všechny tři algoritmy byla zvolena metoda:

```
sample(x : float, y : float) -> float in [0,1]
```

3.2 Pseudokód algoritmů

Value Noise – pseudokód

Algorithm 1 ValueNoise.sample(x, y)

Spočítej celočíselnou souřadnici buňky:
  $x_0 \leftarrow \lfloor x \rfloor$, $y_0 \leftarrow \lfloor y \rfloor$
  $x_1 \leftarrow x_0 + 1$, $y_1 \leftarrow y_0 + 1$

Spočítej lokální souřadnice bodu v buňce:
  $u \leftarrow x - x_0$, $v \leftarrow y - y_0$ ▷ $u, v \in [0, 1)$

Získej náhodné hodnoty v rozích buňky:
  $v_{00} \leftarrow \text{randomValue}(x_0, y_0)$
  $v_{10} \leftarrow \text{randomValue}(x_1, y_0)$
  $v_{01} \leftarrow \text{randomValue}(x_0, y_1)$
  $v_{11} \leftarrow \text{randomValue}(x_1, y_1)$

Spočítej vyhlazené váhy v osách:
  $s_x \leftarrow fade(u)$
  $s_y \leftarrow fade(v)$

Interpoluj v ose x:
  $a \leftarrow lerp(v_{00}, v_{10}, s_x)$
  $b \leftarrow lerp(v_{01}, v_{11}, s_x)$

Výsledná hodnota je interpolace v ose y:
  return $lerp(a, b, s_y)$

Perlin Noise – pseudokód

Algorithm 2 PerlinNoise.sample(x, y)

Spočítej celočíselnou souřadnici buňky:
  $x_0 \leftarrow \lfloor x \rfloor$, $y_0 \leftarrow \lfloor y \rfloor$
  $x_1 \leftarrow x_0 + 1$, $y_1 \leftarrow y_0 + 1$

Spočítej lokální souřadnice bodu v buňce:
  $u \leftarrow x - x_0$, $v \leftarrow y - y_0$

Získej gradientové vektory v rozích buňky:
  $g_{00} \leftarrow \text{randomGradient}(x_0, y_0)$
  $g_{10} \leftarrow \text{randomGradient}(x_1, y_0)$
  $g_{01} \leftarrow \text{randomGradient}(x_0, y_1)$
  $g_{11} \leftarrow \text{randomGradient}(x_1, y_1)$

Spočítej vektory od rohu k bodu:
  $d_{00} \leftarrow (u, v)$
  $d_{10} \leftarrow (u - 1, v)$
  $d_{01} \leftarrow (u, v - 1)$
  $d_{11} \leftarrow (u - 1, v - 1)$

Spočítej skalární součiny (příspěvky z rohů):
  $n_{00} \leftarrow dot(g_{00}, d_{00})$
  $n_{10} \leftarrow dot(g_{10}, d_{10})$
  $n_{01} \leftarrow dot(g_{01}, d_{01})$
  $n_{11} \leftarrow dot(g_{11}, d_{11})$

Spočítej vyhlazené váhy v osách:
  $s_x \leftarrow fade(u)$
  $s_y \leftarrow fade(v)$

Interpoluj příspěvky v ose x:
  $a \leftarrow lerp(n_{00}, n_{10}, s_x)$
  $b \leftarrow lerp(n_{01}, n_{11}, s_x)$

Výsledná hodnota je interpolace v ose y:
  $h \leftarrow lerp(a, b, s_y)$

Normalizuj do intervalu $[0, 1]$:
  return $0.5 \cdot h + 0.5$

Worley Noise – pseudokód

Algorithm 3 WorleyNoise.sample(x, y)

Najdi index buňky, která obsahuje bod (x, y):
  $i_0 \leftarrow \lfloor x \rfloor$, $j_0 \leftarrow \lfloor y \rfloor$

Inicializuj minimální vzdálenost:
  $minDist \leftarrow +\infty$

Pro všechny sousední buňky v okolí 3×3:
for $\Delta j \leftarrow -1$ **to** 1 **do**
  for $\Delta i \leftarrow -1$ **to** 1 **do**
    $i \leftarrow i_0 + \Delta i$
    $j \leftarrow j_0 + \Delta j$
    Získej pozici náhodného feature–pointu v buňce (i, j):
      $(p_x, p_y) \leftarrow$ randomPointInCell(i, j)
    Spočítej eukleidovskou vzdálenost od (x, y):
      $d \leftarrow \sqrt{(p_x - x)^2 + (p_y - y)^2}$
    Aktualizuj minimální vzdálenost:
      $minDist \leftarrow \min(minDist, d)$
  end for
end for

Normalizuj vzdálenost do intervalu $[0, 1]$:
  return normalize$(minDist)$

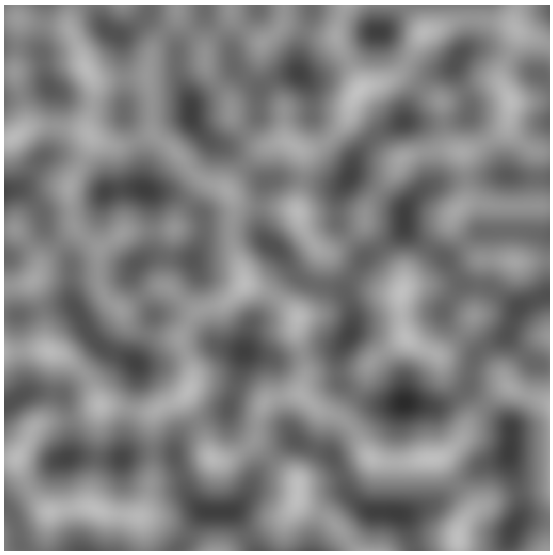
4 Experimenty a výsledky

Experimenty byly provedeny v jazyce Python. Všechny tři noise algoritmy byly testovány se stejnými parametry:

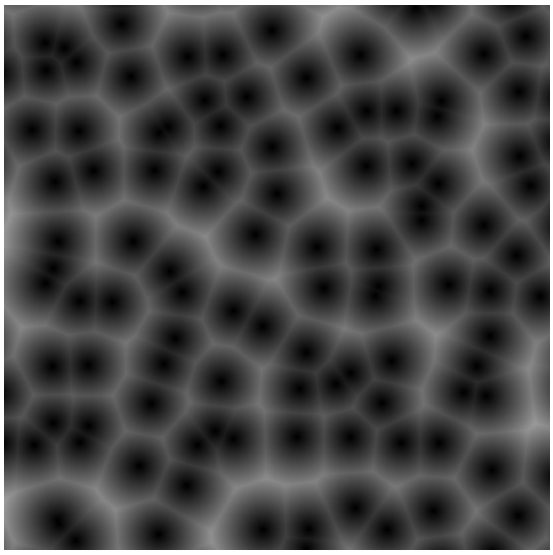
- seed = 1234
- rozlišení 512×512 pixelů
- scale = 0,02

4.1 Výsledné obrázky

Obrázek 1: Value Noise



Obrázek 2: Perlin Noise



Obrázek 3: Worley Noise

4.2 Porovnání

4.2.1 Value Noise

Value noise je vizuálně nejjednodušší ze všech tří testovaných metod. Jeho hlavní vlastností je celková hladkost, která vychází z interpolace náhodných hodnot v rozích buněk. Přechody mezi světlými a tmavými oblastmi jsou poměrně plynulé, nicméně při větším přiblížení je možné pozorovat pravidelnou mřížkovou strukturu. Ta je způsobena tím, že samotné náhodné hodnoty nejsou nijak spojité – pouze interpolované.

Výsledný šum je poměrně „měkký“ a jednoduchý, což může být výhodou nebo nevýhodou v závislosti na aplikaci. Hodí se pro základní generování textur nebo jemné výškové mapy, kde nejsou vyžadovány jemné detaily ani příliš přirozený vzhled. Pro komplexnější struktury je ale Value noise většinou nedostačující.

4.2.2 Perlin Noise

Perlinův šum je z testovaných metod nejvyváženější. Vytváří vizuálně spojitější a hladší struktury než Value noise a zároveň nabízí mnohem přirozenější variabilitu. Díky gradientové konstrukci je šum plynulý ve všech směrech a přechody mezi oblastmi jsou jemné. Perlin noise také nevykazuje výrazné blokové artefakty, které jsou typické pro Value noise.

Jedním z mála možných artefaktů je jemná mřížková stopa při extrémním přiblížení, ale u standardního použití je prakticky neviditelná. Perlinův šum se velmi dobře hodí pro generování terénů, oblaků, organických vzorů nebo přírodních textur. Je to univerzální metoda, která poskytuje realistický, hladký charakter a lze ji snadno kombinovat do více oktáv (fBm), čímž vznikají bohaté a detailní struktury.

4.3 Worley Noise

Worley noise se od předchozích metod zásadně liší. Nevytváří hladké nebo spojité struktury, ale výrazné buněčné vzory připomínající Voroného diagram. Ty jsou charakteristické tmavými oblastmi v místech nejbližších feature pointů a světlejšími přechody směrem k hranicím buněk. Šum nepůsobí hladce, ale spíše ostře a segmentovaně.

Tento typ šumu je velmi vhodný pro efekty, které mají přirozeně buněčný charakter — například textury kamene, popraskaných povrchů, lávy, pěny nebo biomorfních struktur. Naopak se nehodí pro generování terénu nebo měkkých přírodních tvarů, kde je žádoucí spojitost a hladkost.

4.4 Tabulka výsledků

Metoda	Hladkost	Artefakty	Charakter	Typické použití
Value Noise	střední	mřížkové bloky	jednoduchý	základní textury
Perlin Noise	vysoká	minimální	organický	terény, mraky
Worley Noise	nízká	buněčné tvary	segmentovaný	kámen, praskliny

Tabulka 1: Porovnání vlastností jednotlivých noise algoritmů.

5 Závěr

V této práci byly implementovány a porovnány tři algoritmy pro procedurální generování dvourozměrného šumu: Value Noise, Perlin Noise a Worley Noise. Každý z nich představuje odlišný přístup k tvorbě pseudonáhodných struktur a vykazuje specifické vizuální i technické vlastnosti. Implementace byla sjednocena pod jednotné rozhraní a jednotlivé metody byly testovány za stejných podmínek, což umožnilo jejich objektivní porovnání.

Z experimentů vyplynulo, že Value Noise poskytuje nejjednodušší a nejhladší výsledek, avšak jeho vizuální komplexita je omezená a často trpí mřížkovými artefakty. Perlin Noise se ukázal jako nejvyváženější metoda — vytváří spojité, organické struktury s minimem artefaktů a je vhodný pro širokou škálu aplikací, zejména v oblasti počítačové grafiky. Naproti tomu Worley Noise produkuje výrazně odlišné, buněčné a segmentované vzory, které se nehodí pro generování plynulých povrchů, ale jsou ideální pro textury jako kámen, pěna nebo přírodní útvary s ostřejší strukturou.

Celkově lze říci, že každá z metod má své specifické využití. Perlin Noise je univerzální volbou pro přirozeně působící šum, Value Noise se hodí pro jednodušší textury a Worley Noise dominuje při generování buněčných nebo voronoi-jako struktur. Možným rozšířením této práce je implementace více oktáv (fractal Brownian motion), kombinace různých typů šumu které by umožnily vytvářet ještě komplexnější a realističtější procedurální vzory.

Reference

[1] Wikipedia contributors. Perlin noise. Wikipedia, The Free Encyclopedia, 2024. Accessed: 2025-11-28.

[2] Wikipedia contributors. Value noise. Wikipedia, The Free Encyclopedia, 2024. Accessed: 2025-11-28.

[3] Wikipedia contributors. Worley noise. Wikipedia, The Free Encyclopedia, 2024. Accessed: 2025-11-28.