Побудова фрактальних зображень та їх застосування в комп’ютерній графіці

Керівник курсової роботи
Бучко О. А.
Виконала студентка
Косів Х. А.

МЕТА. АКТУАЛЬНІСТЬ. ПОСТАНОВКА ЗАДАЧІ

Мета:

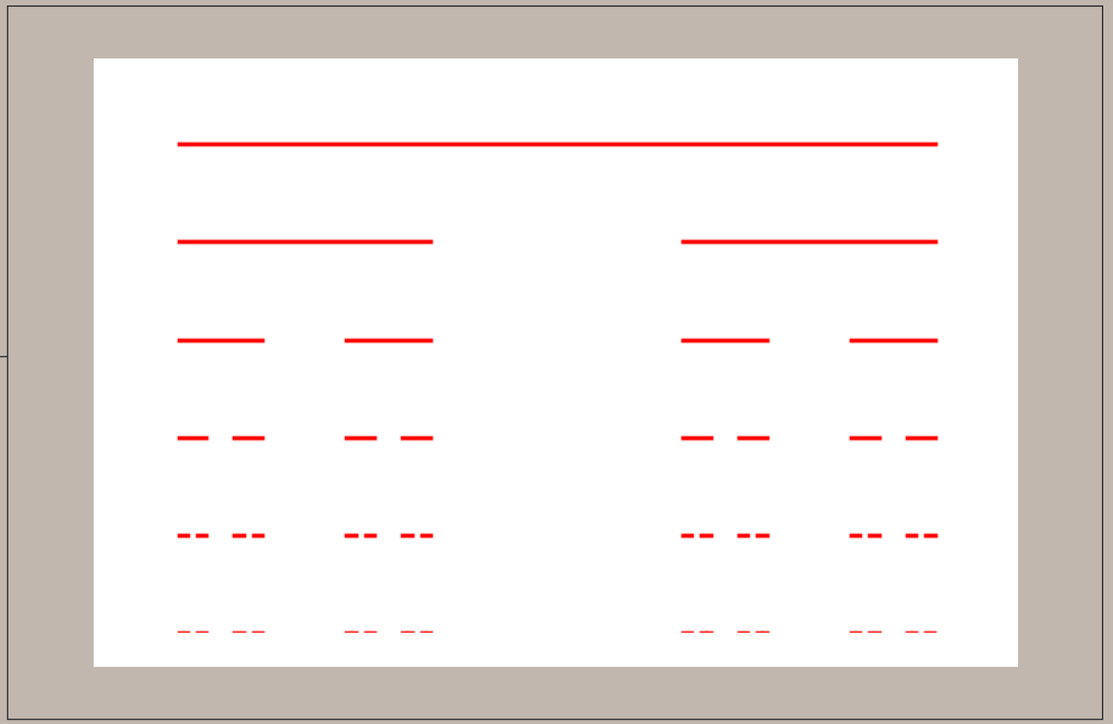
- дослідження математичних основ та практичних аспектів побудови фрактальних зображень, аналіз їх властивостей і візуальних характеристик, а також ознайомлення застосування фракталів у комп’ютерній графіці, моделюванні природних об’єктів.

Актуальність:

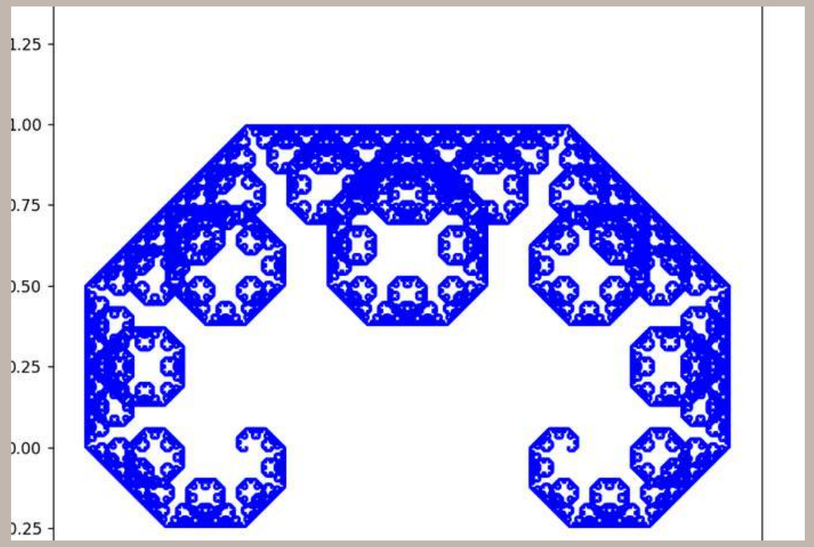
- Завдяки своїй здатності відтворювати самоподібні структури, фрактали знайшли широке застосування у візуалізації, дизайні, цифровому мистецтві, а також у розробці комп’ютерних ігор.

МНОЖИНА КАНТА

```
def cantor_set(ax, x, y, length, depth):
    if depth == 0:
        return
    ax.plot([x, x + length], [y, y], color='red', lw=2)
    l = length / 3
    cantor_set(ax, x, y - 10, l, depth - 1)
    cantor_set(ax, x + 2 *l, y - 10, l, depth - 1)
```



КРИВА ДРАКОНА

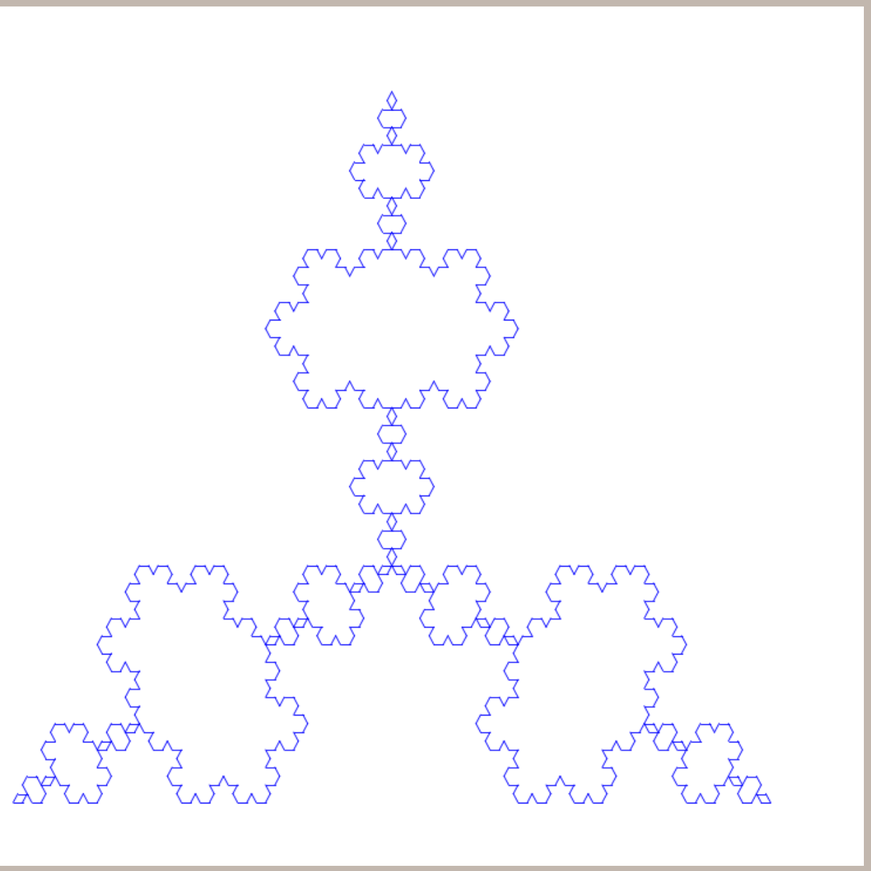


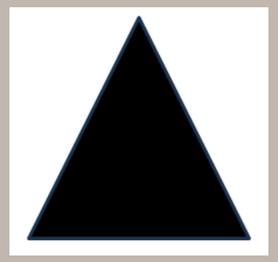
Крива Драковна L-системи	
змінні:	f, h
константи:	+ –
аксіома:	f
правила:	f->f-h h -> f+h
кут:	90 °
крок 1:	f-h
крок 2:	**f-h – f+h**
крок 3:	**f-h – f+h – f-h + f+h**
крок 4:	**f-h – f+h – f-h + f+h – f-h – f+h + f-h + f+h**

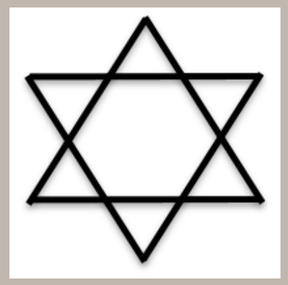
```
def dragon_curve(n_iterations=16):
    points = [(0, 0), (1, 0)]
    for _ in range(n_iterations):
        new_points = [points[0]]
        for i in range(len(points) - 1):
            x1, y1 = points[i]
            x2, y2 = points[i + 1]
            mid_x = (x1 + x2) / 2 + (y1 - y2) / 2
            mid_y = (y1 + y2) / 2 - (x1 - x2) / 2
            new_points.extend([(mid_x, mid_y), points[i + 1]])
        points = new_points
    return points
```

СНІЖИНКА КОХА

```
def koch_snowflake(ax, p1, p2, depth):
    if depth == 0:
        ax.plot([p1[0], p2[0]], [p1[1], p2[1]], color='blue')
        return
 v = (np.array(p2) - np.array(p1)) / 3
 pA = p1 + v
  pB = p1 + 2 *v
angle = np.pi / 3
pC = pA + np.dot([[np.cos(angle), -np.sin(angle)],
[np.sin(angle),  np.cos(angle)]], v)
```







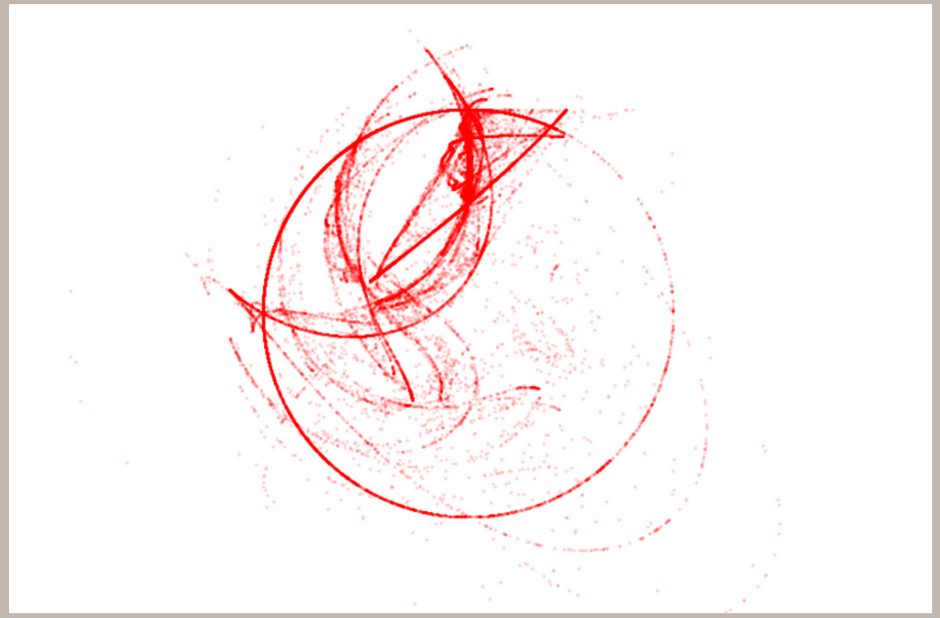
$$N = 3 * (4 ^ N)$$

ЛИСТОК БАРНСЛІ

```
for i in range(1, n_points):
    r = np.random.random()
    if r < 0.01:
        x[i] = 0
        y[i] = 0.16 * y[i - 1]
    elif r < 0.86:
        x[i] = 0.85 * x[i - 1] + 0.04 * y[i - 1]
        y[i] = -0.04 * x[i - 1] + 0.85 * y[i - 1] + 1.6
    elif r < 0.93:
        x[i] = 0.2 * x[i - 1] - 0.26 * y[i - 1]
        y[i] = 0.23 * x[i - 1] + 0.22 * y[i - 1] + 1.6
    else:
        x[i] = -0.15 * x[i - 1] + 0.28 * y[i - 1]
        y[i] = 0.26 * x[i - 1] + 0.24 * y[i - 1] + 0.44
```

a	b	c	d	e	f	p	-
0	0	0	0.16	0	0	0.01	Стебло
0.85	0.04	-0.04	0.85	0	1.60	0.85	Малий листок
0.20	-0.26	0.23	0.22	0	1.60	0.07	Великий листок(ліворуч)
0.15	0.28	0.26	0.24	0	0.44	0.07	Великий листок(праворуч)

ФРАКТАЛЬНИЙ ВОГОНЬ



```
def variation1(x, y): return np.sin(x * y), np.cos(x - y)
def variation2(x, y):
    r = np.sqrt(x**2 + y**2) + 1e-9
    return x / r, y / r
variations = [variation1, variation2, variation3, variation4]
weights = [0.3, 0.3, 0.2, 0.2]
x, y = random.uniform(-1, 1), random.uniform(-1, 1)
for _ in range(50000):
    f = random.choices(variations, weights)[0]
    x, y = f(x, y)
    points.append((x, y))
```

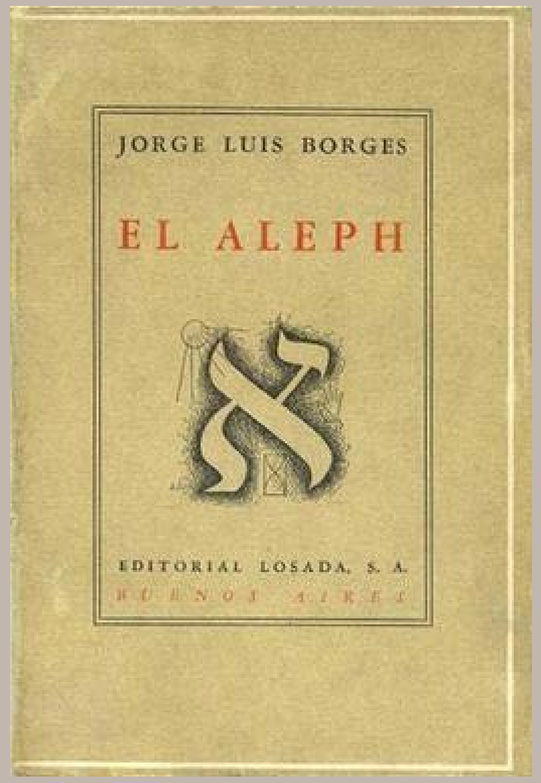
АНАЛІЗ ФРАКТАЛІВ

Назва	Швидкість	Складність	Фрактальна розмірність
Фрактальний вогонь	середня	4.5/5	1.7–2.0
Сніжинка Коха	повільна при багатьох ітерацій	4/5	~1.26
Множина Кантора	дуже швидка	1/5	~0.63
Крива Дракона	середня	4.5/5	~2.0
Листок Барнслі	дуже швидка	5/5	~1.7

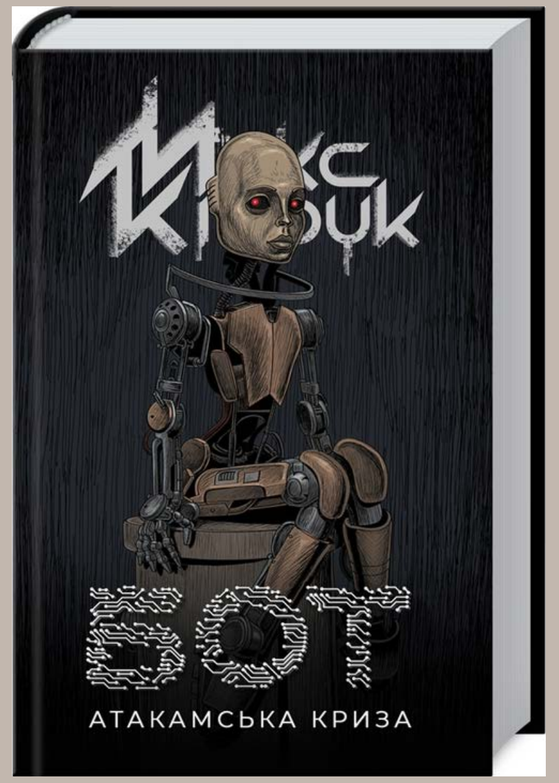
$$D = LOG(N)/(LOG(1/R))$$

ФРАКТАЛЬНІ ЗОБРАЖЕННЯ У РЕАЛЬНОМУ СВІТІ

М. К. ЕШЕР



Х. Л. БОРХЕС



М. І. КІДРУК

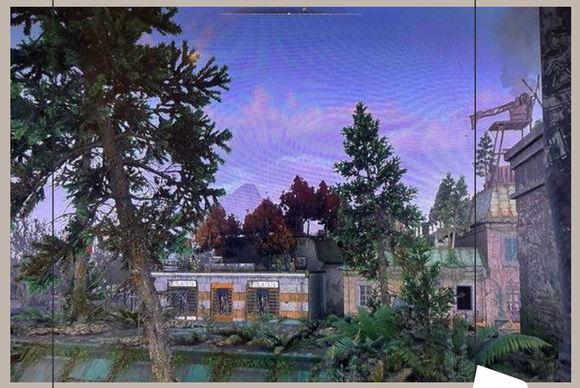
ФРАКТАЛИ

Природна реалізація листка Барнслі навколо.

Days Gone

Ефект и освітлення на мотоциклі, туману створені за допомогою фрактального вогню.

Days Gone



Листок Барнслі, представлений у вигляді папорот і в нижній частині карт инки.

Dying Light 2

Праворуч у центрі видно жовті іскри, які є схожими на фрактальний вогонь.

Dying Light 2

ВИСНОВОК

- Реалізовано побудову п’яти фракталів, описано алгоритм та реалізацію
- Здійснено порівняльний аналіз за трьома критеріями, а саме фрактальна розмірність, швидкість побудови, візуальна складність
- Розглянуто застосування фракталів у реальному світі
- Наведено приклади в ігровій індустрії у природі

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1Fractals in Computer Graphics [Електронний ресурс]. – URL: https://www.geeksforgeeks.org/fractals-incomputer-graphics/

2. Fractal Literature. Yale University. [Електронний ресурс]. – URL: https://gauss.math.yale.edu/fractals/Panorama/Literature/Lit/Literature.html

3. Tessendorf J. Simulating Ocean Water. – Computer Graphics Laboratory, 2005.

4. Hu Y., Velho L., Tong X., Guo B., Shum H. Realistic, Real– Time Rendering of Ocean Waves // Computer Animation and Virtual Worlds. – 2006. – Vol. 17, № 1. – С. 59–67. – DOI: 10.1002.

5. Bird K., Dickerson T., George J. Techniques for Fractal Terrain Generation // HRUMC. – 2013.

6. Belhadj F., Audibert P. Modeling Landscapes with Ridges and Rivers // Proceedings of the 3rd International Conference on Computer Graphics and Interactive Techniques, Australasia and South East Asia. – November 2005. – С. 447–450.

7. Кідрук М. А. Бот : роман / Макс Кідрук. – Харків : Клуб сімейного дозвілля, 2013. – 496 с.

8.Worboys M. F., Duckham M. GIS: A Computing Perspective. – 2nd ed. – Boca Raton : CRC Press, 2004

Дякую за увагу

Керівник курсової роботи

Бучко О. А.

Виконала студентка

Косів Х. А.