

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики
Кафедра інтелектуальних програмних систем

ЗВІТ

з лабораторної роботи

з дисципліни “Математичні основи обчислювальної геометрії”

2D Objects Morphing

Студентів 4 курсу групи ІПС-41

ОС “Бакалавр”

зі спеціальності 121 “Інженерія
програмного забезпечення”

Маліброди Анатолія та

Сукованченко Дмитра

Київ

2022

Постановка задачі

Метою лабораторної роботи є реалізація перетворення одного двовимірного об'єкту в інший з використанням різних часових функцій.

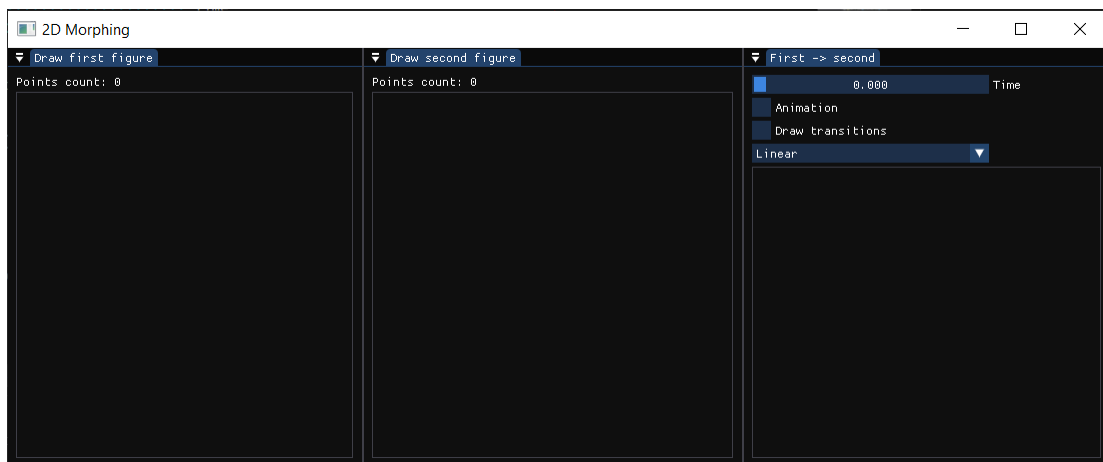
Опис роботи програми

Було реалізовано застосунок, що дозволяє намалювати дві фігури і отримати плавне перетворення між ними, для наглядної демонстрації роботи алгоритму.

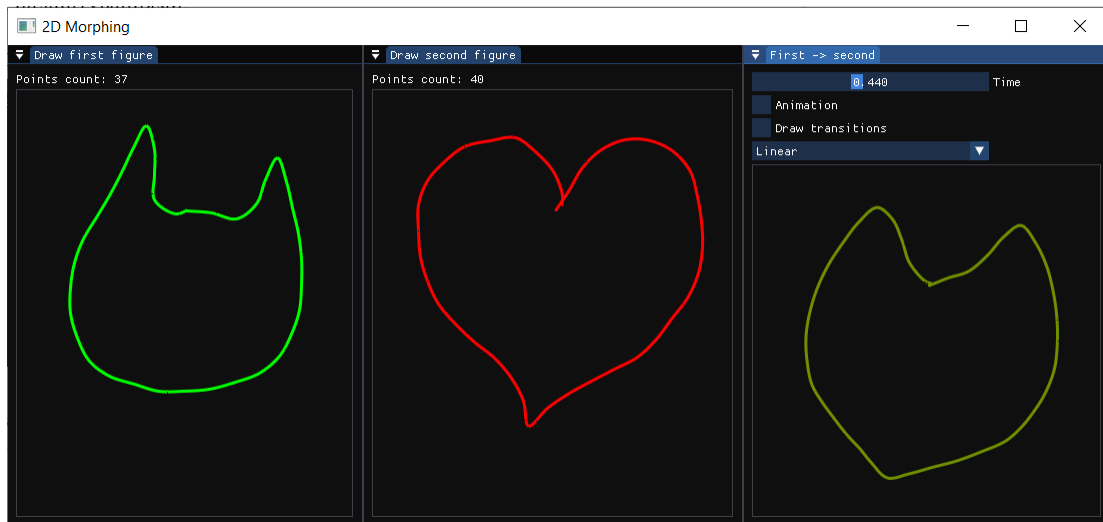
Використані технології: мова програмування C++, фреймворк для розробки інтерфейсу користувача та для малювання фігур ImGui.

Застосунок містить 3 блоки:

- введення вхідної фігури
- введення вихідної фігури
- зображення плавного перетворення із різними налаштуваннями.

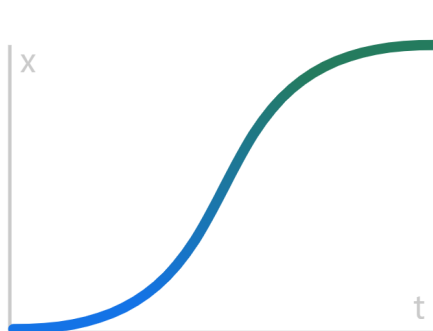


Користувач курсором миші малює дві фігури. Потім у вікні відображення переходу можна подивитись на часову точку переходу t , де $t \in [0, 1]$, яке відповідає лівому зображенню при $t=0$ та відповідно правому при $t=1$. Також для зручності було додано можливість запустити неперервну анімацію перетворення, а також подивитись траєкторію руху точок контуру.

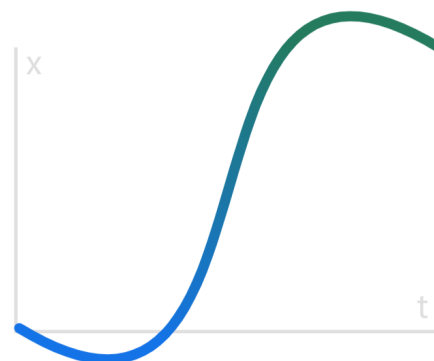


Також крім лінійного перетворення реалізовано анімації із різними часовими функціями, завдяки чому перехід відбувається зі змінною швидкістю, що додає ефекту еластичності.

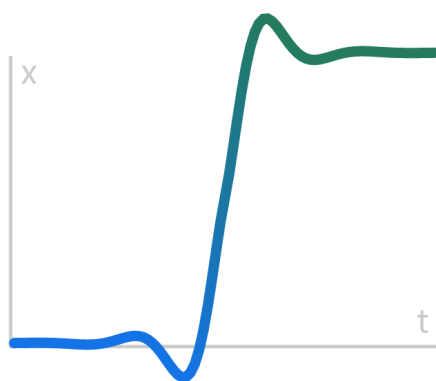
Використані наступні часові функції:



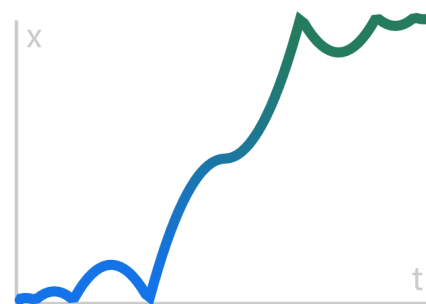
InOutCubic



InOutBack



InOutElastic



InOutBounce

Алгоритм розв'язання

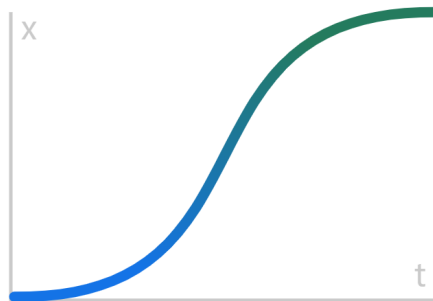
Маємо два контури які між якими потрібно зробити перетворення, для кожної пари точок необхідно задати “траєкторію руху” під час переходу. Нехай А точка на першому зображенні та В точка на другому зображенні, тоді (Ax, Ay) та (Bx, By) відповідні координати цих точок. Для отримання точки перетворення C (Cx, Cy) у момент часу t можна скористатися лінійною формулою $C = (1 - t) * A + t * B$, тобто $Cx = (1 - t) * Ax + t * Bx$ та $Cy = (1 - t) * Ay + t * By$.

Якщо ж маємо різну кількість точок між першим та другим контуром, то додаємо додаткові точки в контур з меншою кількістю доки не зрівняються з іншим контуром. В нашому випадку додаємо точку між випадковими двома іншими сусідніми точками на контурі.

Також розглянути інші часові функції з якими вищезгадана формула набуває вигляду $C = (1 - X) * A + X * B$, де $X = X(t)$. Наприклад:

Кубічна.

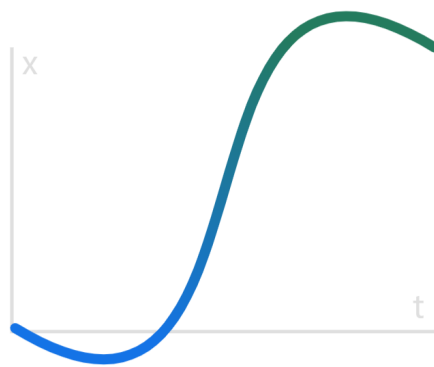
```
function easeInOutCubic(t: number): number {  
    return t < 0.5 ? 4 * t * t * t : 1 - Math.pow(-2 * t + 2, 3) / 2;  
}
```



InOutCubic

Зворотня.

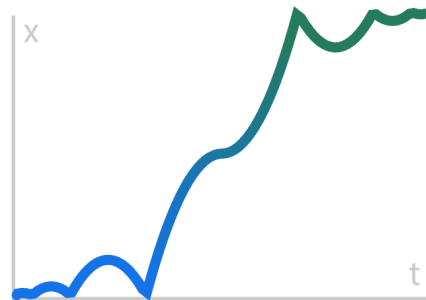
```
function easeInOutBack(t: number): number {  
    const c1 = 1.70158;  
    const c2 = c1 * 1.525;  
  
    return t < 0.5  
        ? (Math.pow(2 * t, 2) * ((c2 + 1) * 2 * t - c2)) / 2  
        : (Math.pow(2 * t - 2, 2) * ((c2 + 1) * (t * 2 - 2) + c2) + 2) / 2;  
}
```



InOutBack

З відскоком

```
function easeInOutBounce(t: number): number {
  return t < 0.5
    ? (1 - easeOutBounce(1 - 2 * t)) / 2
    : (1 + easeOutBounce(2 * t - 1)) / 2;
}
```

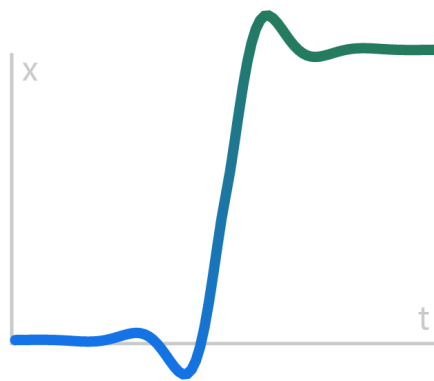


InOutBounce

Еластична

```
function easeInOutElastic(t: number): number {
  const c5 = (2 * Math.PI) / 4.5;

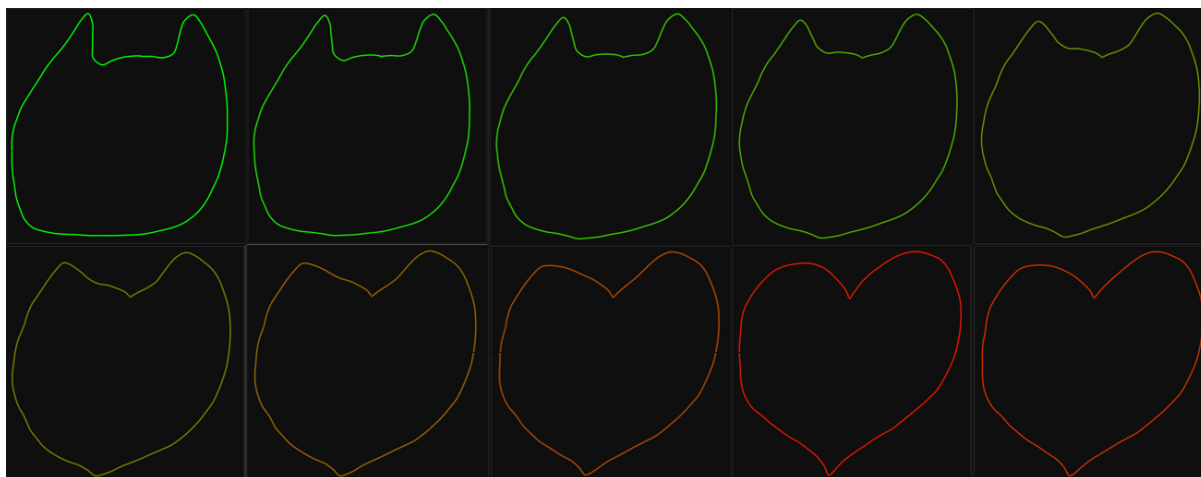
  return t === 0
    ? 0
    : t === 1
    ? 1
    : t < 0.5
    ? -(Math.pow(2, 20 * t - 10) * Math.sin((20 * t - 11.125) * c5)) / 2
    : (Math.pow(2, -20 * t + 10) * Math.sin((20 * t - 11.125) * c5)) / 2 + 1;
}
```



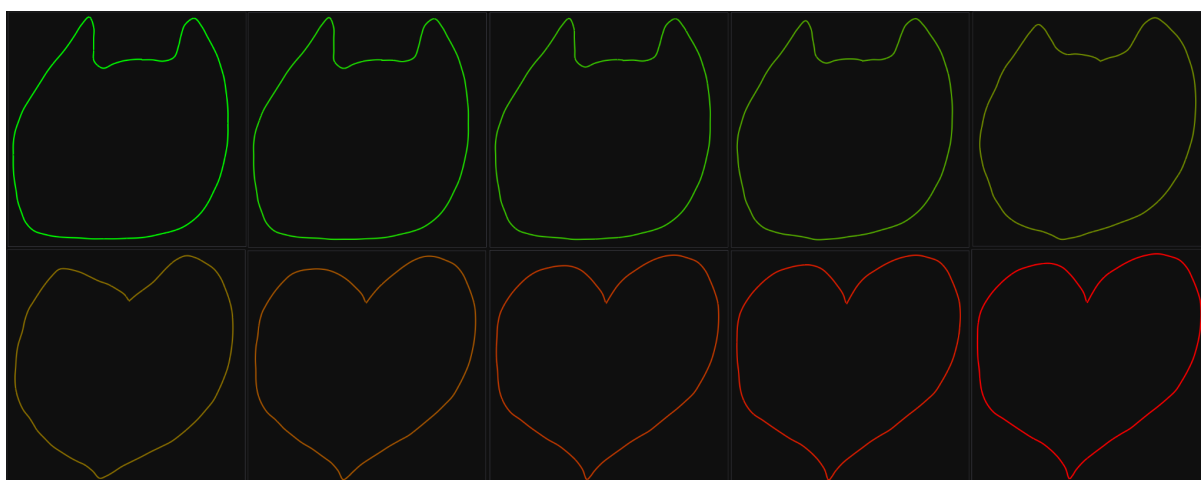
InOutElastic

При малюванні контуру точки беруться з невеликим інтервалом та по ним будується сплайн для кращого сприйняття малюнку.

Демонстрація реалізації



Лінійна



Кубічна

Детальне відео-демонстрацію можна подивитись за посиланням:
<https://drive.google.com/file/d/18CbNQazGTrSJtqzGCBGwhppefa3vWGTj/view?usp=sharing>

Код програми можна подивитись за посиланням:
https://github.com/antl-m/moog_tasks/tree/main/lab

Висновки

В результаті виконання лабораторної роботи було реалізовано алгоритм для перетворення. Також продемонстровано при роботі з лінійною функцією часу, кубічною, зворотною, з відскоком та еластичною.

Морфінг двовимірних об'єктів досить розповсюджений у медіа сфері, часто можна побачити його використання у різних рекламних роликах або ж як перехід між різними фрагментами у відео. Також деякі комп'ютерні ігри використовують цей прийом.

Список використаної літератури

1. <https://www.sciencedirect.com/science/article/pii/S1631073X16300802>
2. <https://en.wikipedia.org/wiki/Morphing>
3. <https://davis.wpi.edu/~matt/courses/morph/2d.htm>
4. <https://www.101computing.net/python-turtle-morphing-algorithm/>