

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ІВАНА ФРАНКА
Факультет прикладної математики та інформатики

Кафедра дискретного аналізу

Обробка зображень і мультимедіа
Індивідуальне завдання №3

Виконала:

Студентка групи

ПМі-43

Шувар Софія

Викладач:

Гутік О.В.

Хід роботи

1. Програмна реалізація виконана на мові програмування Python з використанням можливостей бібліотеки Pillow для створення зображень та imageio для створення файлу формату .gif.
2. В результаті роботи створила .gif файл - банер популяризації спеціальності Комп'ютерні науки. Розмір файлу 500*125 пікселів. Файл містить текст, що рухається вгору та вниз, кільце, що змінює колір та розмір, графічне зображення ноутбука та кілька миготливих зірок.



3. Для створення були розроблені 60 кадрів анімації, у цикл довжиною 60 разів. У кожній ітерації циклу:
 - нове зображення створюється з чорним фоном.
 - Текст "COMPUTER SCIENCE" малюється у відповідному положенні.
 - Кільце малюється конкретного радіусу та заповнюється певним кольором.
 - Графічне зображення ноутбука малюється в центрі кільця.
 - Створюється список положень зірочок, і цикл проходить по цих позиціях щоб намалювати зірки. Зірки налаштовано на мигання, вони змінюють свій колір через кожний другий кадр у циклі.

По завершенню циклу кадри збираються в список, і зберігаються як анімований .gif файл із частотою кадрів 30 кадрів/с.

Далі більш детально буде висвітлено процес створення кожної компоненти.

4. **Текст.** Завантажую шрифт із файлу формату .ttf попередньо викаченого з інтернету. Використовую шрифт DEBROSEE. Розмір шрифту визначаю як третину висоти банера.



Координату x - початкової точки для малювання тексту визначаю як ширину зображення мінус ширину тексту мінус 20 пікселів для деякого відступу. у-координату початкової точки для малювання тексту обчислюю як $y = 50 + 20 * \text{math.sin}(i / 5)$, де i це номер ітерації. Це створює хвилеподібний рух тексту. Колір тексту встановлюю як білий - (255, 255, 255).

5. **Кільце.** Малюю кільце на фоновому зображенні за допомогою методу `draw.ellipse` з бібліотеки `Pillow`. Кільце розташоване вертикально по центру зображення та розташоване на фіксованій відстані від лівого краю. Розмір кільця змінюється з часом відповідно до синусоїди, яка контролюється змінною `ring_radius = 60 + 10 * \text{sin}(i/10)`, таким чином радіус кільця коливається від 60 до 80 пікселів.

Колір кільця також змінюється з часом, змінюючи діапазон кольорів. Змінна `ring_color_step` контролює розмір кроку між кольорами. У цьому випадку колір кільця змінюється на $255 // \text{num_circles}$ (8) кожного кадру, що означає, що для циклічного переходу між усіма кольорами потрібно 30 кадрів. Фактичний колір обчислюється за допомогою змінної i , яка є поточним номером кадру, і операції за модулем для обтікання діапазону кольорів.

Позиція x обчислюється так, що лівий край кільця знаходиться на відстані 60 пікселів від лівого краю зображення. Позиція y розраховується таким чином, щоб кільце було відцентровано вертикально на зображенні. Товщину контуру кільця встановила 4 пікселі та заповнила її блим кольором.

6. **Зображення ноутбука.** Зображення ноутбука з прозорим фоном створила використовуючи платформу Canva.



Спочатку зображення завантажується за допомогою методу `Image.open` і перетворюється в режим `RGBA` за допомогою методу `convert`. Режим `RGBA` включає альфа-канал, який забезпечує прозорість.

Далі за допомогою методу зміни розміру регулюється розмір зображення ноутбука. Ширина та висота зображення встановлюються рівними подвоєному значенню початкового значення змінної $ging_radius = 50$, яка визначає розмір кільця.

У циклі обчислюється його положення в кільці. Обчислюю координати x і y , відповідно, верхнього лівого кута зображення ноутбука. Властивості ширини та висоти зображення ноутбука використовуються для центрування зображення в кільці.

Зображення ноутбука вставляється на поточний кадр за допомогою методу вставки. Параметр маски встановлюється на альфа-канал зображення ноутбука за допомогою методу розділення. Це гарантує, що зображення ноутбука плавно зливається з фоном.

7. **Мигаючі зірки.** Створила список `star_positions` на обраній відстані між зірками = 30. Потім створюється екземпляр `ImageDraw.Draw()` під назвою `draw` для малювання на зображенні та циклічно переглядає кожну позицію в `star_positions`. Якщо індекс k позиції при діленні на 3 дає такуж остачу як поточним значенням i поділене на 3 (із зовнішнього циклу), зірка матиме жовтий колір (255, 255, 0). Якщо ця різниця рівна 1, то оранжевий (255, 165, 0) в іншому випадку - білий. (255, 255, 255). Таким чином будь які 3 зірки, що стоять поруч матимуть різні кольори, а їх кольори зсуватимуться вправо на 1 при кожній зміні кадру.

Потім викликається функція `draw_star()`, що використовується для обчислення позицій вершин п'ятикутної зірки, які потім використовуються для малювання зірки. Положення вершини зірки обчислюються за допомогою тригонометрії та циклу, який повторюється п'ять разів. Змінна циклу i представляє поточну ітерацію та має значення від 0 до 4 (включно). У середині циклу кут (у градусах) для кожної вершини обчислюється за формулою $angle_deg = 36 + i * 72$. Це гарантує, що кути рівномірно розподілені навколо центру зірки. Далі кут перетворюється в радіани за формулою $angle_rad = \text{math.pi} / 180 * angle_deg$. Координата x -вершини обчислюється за формулою $x + radius * \text{math.cos}(angle_rad)$, де x - координата x центру зірки, $radius$ - радіус кола, у яке вписано зірку. Координата y обчислюється аналогічно за формулою $y - radius * \text{math.sin}(angle_rad)$. Для точок внутрішньої частини зірки радіус ділиться на 2, а кут збільшується на $\text{math.pi} / 5$ (що дорівнює 36 градусам), щоб забезпечити правильний кут внутрішніх точок. Отримані координати використовується для малювання зірки за допомогою методу `draw.polygon`.

8. Таким чином утворилось 60 зображень.



9. Щоб зберегти послідовність кадрів як анімований GIF використовую бібліотеку `imageio`, а саме функцію `imageio.mimsave()`, вказавши частоту кадрів в секунду - 30. Для .gif файлу використовується кодування у форматі Graphics Interchange Format (GIF), яке використовує стиснення без втрат для зменшення розміру файлу анімації. Цей формат зазвичай використовується для невеликих анімацій і підтримується більшістю веб-браузерів.

10. Також я розмістила створений анімований файл на веб-сторінці.

Банер популяризації спеціальності Комп'ютерні науки



```
Elements Console Sources Network Performance >> ⚙️
***<doctype html> == $0
<html>
  <head>
    <title>Банер популяризації спеціальності Комп'ютерні науки</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <div style="text-align:center;">
      <h1>Банер популяризації спеціальності Комп'ютерні науки</h1>
      
    </div>
  </body>
</html>
```