PROCESARE DE IMAGINE

ÎN JAVASCRIPT

Proiect realizat de:

Sătmar Elena – Elisabeta

Grupa 323AC

Cuprins

1. Descrierea aplicației cerute
2. Teorie
3. Descriere implementare
4. Descrierea funcțională a aplicației
5. Bibliografie

# Descrierea aplicației cerute

Aplicația va lua o dată la 7 secunde câte o imagine în mod aleator de pe site-ul <https://dog.ceo/api/breeds/image/random> și o va recrea în aplicație. După o secundă de la apariția imaginii originale, o nouă imagine va apărea în paralel cu cea originală. Asupra acesteia se vor aplica în mod consecutiv, la o distanță de 1 secundă procesul de oglindire, urmat de cel de transformare a culorilor în alb-negru folosind metoda ponderată. Astfel, clientul poate observa cele 2 imagini în pararel, atât cea originală, cât și cea procesată de către aplicație. După cele 7 secunde, aplicația își dă refresh și o nouă imagine va fi procesată.

# Teorie

JavaScript este un limbaj de programare folosit pentru a adăuga funcționalități interactive unui site web. În cazul procesării imaginilor, putem utiliza JavaScript pentru a manipula și modifica imagini încărcate în pagina web.

Un mod comun de a face acest lucru este prin intermediul unui element canvas HTML. Acesta este un element de tip bloc care poate fi utilizat pentru a desena grafice, imagini sau alte elemente prin intermediul unui context 2D sau 3D. Pentru a procesa o imagine cu ajutorul canvas-ului, putem încărca imaginea într-un element canvas, utilizând apoi metodele contextului 2D pentru a modifica imaginea.

De exemplu, putem folosi metoda drawImage() pentru a desena o imagine în canvas și apoi putem utiliza metode precum getImageData() și putImageData() pentru a accesa și modifica pixelii individuali ai imaginii. Putem de asemenea utiliza alte metode pentru a aplica filtre sau transformări imagini înainte de a fi afișate în canvas.

În plus, există multe biblioteci JavaScript care oferă suport pentru procesarea imaginilor și care pot face acest lucru mai ușor, precum și oferind mai multe opțiuni și funcționalități.

În mod obișnuit, o imagine în culori este reprezentată ca o matrice de pixeli, fiecare pixel conținând valori pentru culorile roșu, verde și albastru (RGB). În cazul unei imagini în tonuri de gri, fiecare pixel conține o singură valoare care reprezintă intensitatea griului pentru acel pixel.

Un mod comun de a converti o imagine în tonuri de gri este prin utilizarea metodei "ponderată", în care se atribuie fiecărei culori o pondere diferită în funcție de importanța sa în perceperea culorii finale de către ochiul uman. De exemplu, se poate asuma că ochii umani sunt mai sensibili la culoarea verde decât la celelalte două culori, astfel încât valoarea pentru culoarea verde poate fi ponderată mai mare în calculul tonului de gri final.

# Descriere implementare

Crearea unui html unde se vor afișa informațiile și link-uirea acestuia la un css și javascript. Crearea unui mic canvas în care se va procesa imaginea. Crearea unei funcții care proceseaza fiecare pixel al imaginii și îl transformă în nuanțe de alb, negru și gri. Crearea unei funcții care preia imaginea de pe site-ul dorit. Crearea unei funcții care ajută la afișarea imaginii în aplicație și la reîncărcarea acesteia după un timp prestabilit.

# Descriere funcțională a aplicației

Funcție: Încărcare Imagine

Intrare: Fișier imagine selectat de către utilizator

Ieșire: Imagine afișată în aplicație

Funcție: Aplicare Filtru

Intrare: Tipul filtrului și parametrii selectați de către utilizator

Ieșire: Imagine filtrată afișată în aplicație

Funcție: Salvare Imagine

Intrare: Fișier imagine și locația de salvare selectate de către utilizator

Ieșire: Imagine salvată în locația specificată

# Bibliografie

* <https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API>
* <https://reactjs.org/>
* <https://angular.io/>
* <https://eloquentjavascript.net/>
* <https://www.w3schools.com/>