

Toteutusdokumentti

Ohjelman yleisrakenne

Ohjelma koostuu Audioprocessing-luokasta ja yksinkertaisesta käyttöliittymästä. Audioprocessing-luokka, joka sisältää nopean fourier muunnoksen, äänisuodattimet sekä äänentoistamisen logiikan. Käyttöliittymään taas liittyvät play_signal ja play_audio, jotka valitsevat oikean toistettavan audiotyyppin ja toistavat sen.

Aika- ja tilavaativuudet

Cooley Tukeyn aikavaativuus on $N \log N$ ja tilavaativuus $O(N)$, jossa N on näytteiden lukumäärä. Ohjelman heikkous on kyllä siinä, että se suorittaa kaikki äänisuodatintyytit kerralla ohjelman alussa. Tästä johtuen ohjelma on nelinkertaisesti hitaampi $4 \times N \log N$.

Työn mahdolliset puutteet ja parannusehdotukset

Haluaisin lisätä ohjelmaan "stop"-napin. Haluaisin myös, että kunkin äänensuodattimen analyysit pyöritettäisiin erikseen, jotta alussa odotusaika olisi nopeampi. Visualisointikin jäi toteuttamatta.

Laajojen kielimallien käyttö

Käytin Chatgpt:tä apuna työn ideoimisessa: Mitä työkaluja ja kirjastoja tarvitsen ja voisin käyttää apuna projektissa. Missä järjestyksessä ohjelman kannattaisi tehdä asioita ja mitä konkreettisia vaiheita minulla on, joita varsinaisen ohjelmarakenteen pitäisi sisältää. Lisäksi käytin Chatgpt:tä apuna yksikkötestien ideoimisessa, tosin monet ideoista eivät olleet kauhean hyviä, mutta saattoivat johtaa parempaan ideaan.

Lähteet

Yliluoma, J. (2024). *Nopea Fourier-muunnos – Teoria ja toteutus modernilla C++:lla*. Helsingin yliopisto

[Cooley–Tukey FFT algorithm \(Wikipedia\)](#)

Videot

[Simon Xu: Discrete Fourier Transform - Simple Step by Step](#)

[MATLAB: Understanding the Discrete Fourier Transform and the FFT](#)

[Reducible: The Fast Fourier Transform \(FFT\): Most Ingenious Algorithm Ever?](#)

[Simon XU: The FFT Algorithm - Simple Step by Step](#)

