

GYMNÁZIUM JÁNA ADAMA RAYMANA

AUDIO VISUALIZER

ROČNÍKOVÁ PRÁCA

Autor: Ema TUPTOVÁ

Prešov 2023

Abstrakt

Technologický pokrok ľudstvu pomáha a uľahčuje našu každodennú činnosť. Navyše, umožňuje vytvárať a prezentovať umenie novými spôsobmi, experimentovať. Spojenie technológií a umenia sa každým dňom spevňuje. Výsledkom čoho sú rôzne interaktívne prvky v galériách, múzeách, na festivaloch či DJ setoch. Táto ročníková práca bola vytvorená so zámerom prepojenia práve hudby a programovania, a to pomocou programovacích jazykov HTML, CSS a JavaScript. Program generuje animované obrazy v reálnom čase z dát, ktoré získava z mikrofónu zariadenia. Spracúva vlastnosti zvuku a premieta ich, čo najpresnejšie, jednoduchým efektom na obrazovku. Ďalej ponúka užívateľovi rôzne nastavenia prostredia podľa jeho predstáv. Výsledkom je atraktívna a funkčná aplikácia. Tento projekt mi umožnil prehĺbiť moje znalosti v danej téme a bol pre mňa zaujímavou skúsenosťou pri práci s programovacím jazykom JavaScript. Celkový výsledok bol úspešný. V budúcnosti by mohla byť práca použitá ako základ pre moje projekty s podobným zámerom.

Kľúčové slová: audio, hudba, zvuk, vizualizácia, zvukový signál, mikrofón, animácia, programovanie, vykresľovanie v reálnom čase, HTML, CSS, JavaScript.

Prehlásenie

Prehlasujem, že som túto prácu vypracovala samostatne a že som uviedla všetku použitú literatúru a všetky použité informačné zdroje.

Obsah

Úvod	4
1 Teoretická časť	5
1.1 Zvuk vo fyzike	5
1.2 Zvuk v informatike	5
1.3 Audio vizualizér.....	6
1.4 Použité programovacie jazyky.....	6
2 Praktická časť	7
2.1 Spracovanie zvuku	7
2.2 Vizualizácia a animácia	8
2.2.1 Prostredie	8
2.2.2 Animácia.....	8
2.2.3 Objekty a ich správanie	8
2.3 Interaktívne funkcie	9
2.3.1 Color Picker	9
2.3.2 Veľkosť objektov.....	9
2.3.3 Party Mode.....	10
2.3.4 Spotify playlist.....	11
Záver	12
Zoznam bibliografických odkazov	13
Zoznam príloh.....	14

Úvod

Výraz „*the art of coding*“, teda v preklade krása programovania, sa stáva v súčasnosti čoraz viac používaným. Programovanie a umenie majú viac spoločného, ako to môže na prvý pohľad vyzerat'. Nie sú to iba mechanické, ale najmä kreatívne procesy, pri ktorých sa neustále učíme. Tvoríme, robíme chyby a zdokonaľujeme sa. Taktiež, počítačový program nemusí byť iba nápomocný, no aj zábavný, inšpiratívny či esteticky príjemný.

Pri výbere témy na túto ročníkovú prácu sme teda spojili dve záľuby – hudbu a programovanie. Jej výsledkom je náš audio vizualizér, ktorý dokáže zachytiť nehmateľný zvukový signál a preniesť jeho vlastnosti pomocou grafických prvkov v reálnom čase na monitor. V našom prípade bolo cieľom za pomoci HTML, CSS a JavaScript vytvoriť program, ktorý bude reagovať na zvuky okolia, vizuálne doplní hudobný zážitok a vytvorí pulzujúcu atmosféru na základe požiadaviek užívateľa, čo sa nám aj podarilo. Vytváranie vizualizéru tiež odhalilo, že programovanie môže byť zábavné a kreatívne – skutočné umenie.

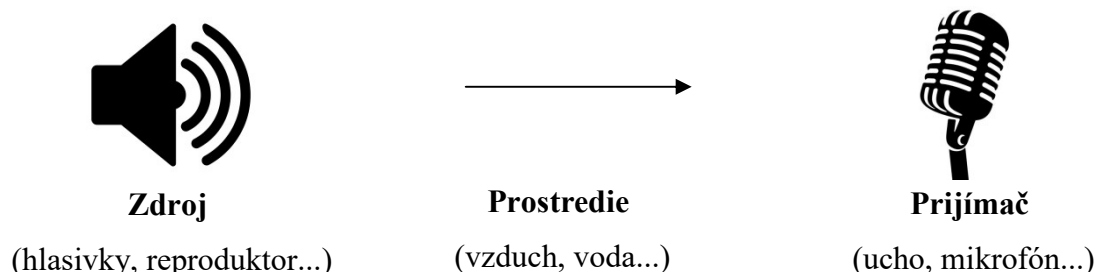
Výsledný audio vizualizér umožňuje vytvoriť vizuálnu predstavu o prehrávanej hudbe, a taktiež vidieť zvuky, ktoré inak zostávajú skryté. Používanie našej aplikácie sa stalo skutočným zážitkom, pretože vizuálne efekty boli plne prispôsobiteľné a mohli sme ich upraviť podľa našich preferencií. V nasledujúcich častiach ročníkovej práce, ktorá je rozdelená na teoretickú a praktickú časť, popisujeme ako sme tento audio vizualizér vytvárali.

1 Teoretická časť

Na vytvorenie pútavého zážitku (vizualizácia zvukových signálov ako grafické prvky na obrazovke) je potrebné ovládať základné poznatky o zvuku a až následne ich prepojiť s programovaním. Slúži na dorozumievanie sa, orientáciu, varovanie pred nebezpečenstvom, signalizáciu, terapiu, či zábavu...

1.1 Zvuk vo fyzike

Zvuk je mechanické vlnenie, ktoré pôsobí na ľudské ucho a vyvoláva v ňom sluchový vnem (frekvencia 16 Hz - 20000 Hz). Šíri sa prostredím v podobe tlakových vln.



Základné vlastnosti zvuku: výška (udáva frekvencia), hlasitosť (udáva amplitúda), farba tónu (udáva zdroj).

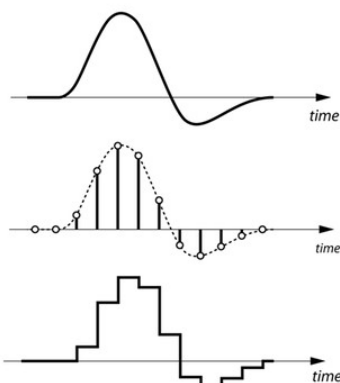
1.2 Zvuk v informatike

Z informatického hľadiska sa zvuk skladá z digitálneho signálu (0/1), ktorý reprezentuje zmeny tlaku v čase. Pre záznam zvuku sa využívajú mikrofón, zvuková karta a softvér, ktoré digitalizujú analógový signál. Tento proces prebieha v dvoch krokoch:

1. Vzorkovanie – záznam hodnôt analógového signálu v pravidelných intervaloch s určitou frekvenciou. Bežne sa používa 22 050 Hz a 44 100 Hz.
2. Kvantovanie – zaokrúhľovanie nameraných hodnôt na najbližšie použiteľné hodnoty. Hodnota kvantovania ovplyvňuje kvalitu zvuku (viac hodnôt pre reprezentáciu zvukového signálu = vyššia kvalita). Bežne sa používa 16 a 32 bitov.

Následne môžeme tieto digitálne hodnoty uložiť alebo prenášať digitálnymi médiami.

Naopak, pre reprodukciu zvuku sa využívajú reproduktory, ktoré prevádzajú digitálny signál na analógový a vytvárajú zvuk.



Analógový signál

Digitalizácia

Digitálny signál

1.3 Audio vizualizér

Účelom efektívnej vizualizácie hudby je čo najlepšie synchronizovať charakteristiky hudobnej stopy a objekty vizuálneho obrazu, ktoré sa vykresľujú a zobrazujú v reálnom čase. Existuje niekoľko foriem audio vizualizérov, od tých jednoduchších, ktoré sme sa snažili vytvoriť, až po komplexné 3D vizualizácie využívané hudobníkmi, zvukármi či zvukovými inžiniermi.



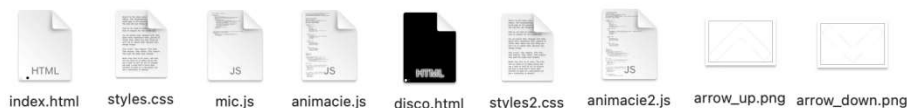
1976 – Atari Video Music, prvý audio vizualizér

1.4 Použité programovacie jazyky

HTML (HyperText Markup Language) je používaný na tvorbu webových stránok. HTML umožňuje vytvárať stránky obsahujúce text, obrázky, videá a iné prvky.

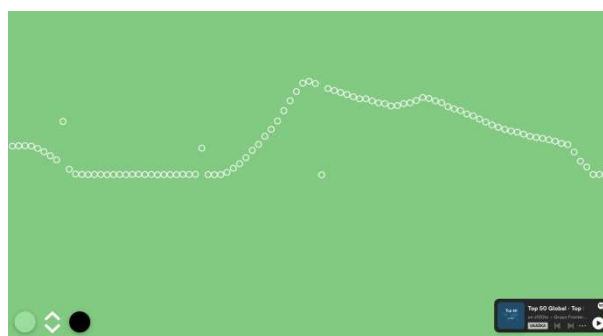
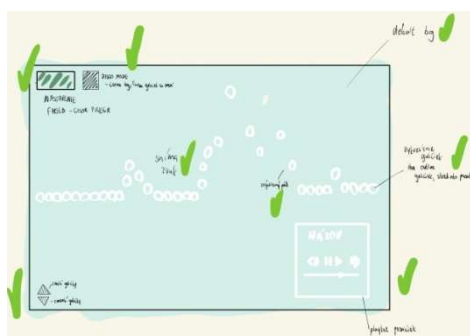
CSS (Cascading Style Sheets) je programovací jazyk, ktorý sa používa na definovanie vzhľadu webových stránok v HTML. Pomocou CSS sa určujú farby, fonty, rozmery, pozície a mnoho ďalších parametrov, ktoré ovplyvňujú vzhľad webovej stránky.

JavaScript umožňuje vytvárať interaktívne prvky na webových stránkach. Používa sa na tvorbu funkcií, ktoré môžu reagovať na používateľské akcie. JavaScript umožňuje dynamickú zmenu obsahu stránky bez nutnosti jej opätovného načítania.



2 Praktická časť

Samotné programovanie projektu bolo dlhým procesom, neustálym hľadaním chýb a ich následným vylepšovaním. Na začiatku sme vytvorili jednoduchý náčrt programu, v ktorom sme postupne plnili úlohy a dopĺňali nové pokiaľ audio vizualizér nadobudol finálnu podobu.

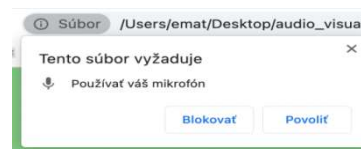


Porovnanie to-do listu a finálnej verzie programu

2.1 Spracovanie zvuku

Zvuky a hudbu z okolia je potrebné previesť na digitálny signál. Používajú sa na to rôzne technológie a postupy. V tejto práci sme využili spracovaný kód v prostredí JavaScript. Ten je následne prepojený s HTML súborom ako script. Umožňuje rýchle a jednoduché spracovanie zvukových dát. Ďalej ho využívame už na vytvorenie samotnej animácie skákajúcich guľôčok. V našom prípade sme museli hodnoty vložiť do absolútnej hodnoty (iba kladné čísla) a vynásobiť ich číslom 7 (viditeľnejšie zmeny).

Kód na získanie dát z mikrofónu zariadenia



```
1 class Microphone {
2   constructor () {
3     this.initialized = false;
4     navigator.mediaDevices.getUserMedia({audio: true})
5     .then(function(stream) {
6       this.audioContext = new AudioContext();
7       this.mic = this.audioContext.createMediaStreamSource(stream);
8       this.analyzer = this.audioContext.createAnalyser();
9       this.analyzer.fftSize = 1024;
10      const bufferLength = this.analyzer.frequencyBinCount;
11      this.dataArray = new Uint8Array(bufferLength);
12      this.mic.connect(this.analyzer);
13      this.initialized = true;
14    }).bind(this)).catch(function(err) {
15      alert(err);
16    });
17  }
18
19  getSamples() {
20    this.analyzer.getByteTimeDomainData(this.dataArray);
21    let normSamples = [...this.dataArray].map(e=> e/128 - 1);
22    return normSamples;
23  }
24 }
```

← pýta si povolenie používať mikrofón

← ak povolené

← dĺžka array = fftSize/2

← ak zamietnuté

← usporiadané dáta pripravené na použitie

2.2 Vizualizácia a animácia

2.2.1 Prostredie

V úvode sme si vytvorili prázdny dokument HTML, ktorý sme nazvali *Audio Visualizer ET*. Následne sme ho prepojili s CSS súborom, kde sme nastavili základné vlastnosti pre celé telo dokumentu. Snažili sme sa vytvoriť jednoduchý funkčný dizajn. Nebol to problém, keďže sme už s týmito jazykmi pracovali v škole na hodinách informatiky pri vytváraní našich profesijných webstránok.

2.2.2 Animácia

Animáciu objektov sme vytvárali pomocou JavaScript. Aby sme vykreslili zvuk na obrazovke, museli sme vytvoriť canvas element. Canvas je virtuálne plátno, na ktoré sa vykresľujú rôzne prvky, objekty, texty... Ďalej sme zadefinovali CTX (context), pomocou ktorého vieme vykresľovať objekty ako kruhy, obdĺžniky, krivky.

2.2.3 Objekty a ich správanie

Na reprezentáciu zvuku sme si vybrali skákajúce guľôčky. Vytvorili sme triedu Dot, ktorá potrebuje tri parametre, a to hodnotu x, hodnotu y a polomer. Ďalej sme vytvorili funkciu vykreslenia pomocou ctx, kde sa kruh vykresľuje v radiánoch.

Keďže sme sa pokúšali o čo najlepší zážitok z aplikácie, bolo potrebné nastaviť responzívne generovanie guľôčok v závislosti od šírky canvasu. Vytvorili sme preto for i loop, ktorý vykresľoval guľôčky zaradom, kým bol na canvase priestor. Aby užívateľ nemusel refreshnúť stránku pri každej zmene, nastavili sme automatický reload stránky vždy, keď sa zmení šírka okna.

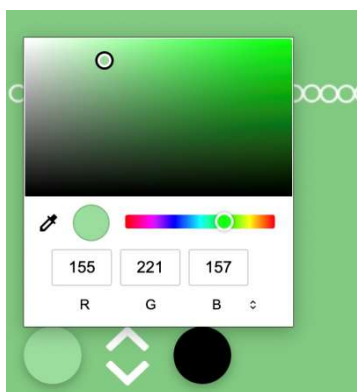
Samotný pohyb sme riešili vo funkcii movement(), kde sme využili niekoľko podmienok a metódu requestAnimationFrame(movement), vďaka ktorej sa v každom frame vyčistí celý canvas a vykreslí opäť s novými hodnotami. Definovali sme, ako majú guľôčky padat' (zrýchlený pohyb) a ako majú skákať (spomalený pohyb), aby to vyzeralo reálne, čo si vyžadovalo mnoho pokusov. Vytvorili sme slučku, kde guľôčky padajú, až kým sa nedostanú do polovice výšky canvasu. Potom je každej guľôčke priradená istá nameraná hodnota mikrofónu, o ktorú sa zmenší hodnota y, a potom začne opäť padat'. Na tomto princípe funguje naša animácia reagujúca na zvuk.

2.3 Interaktívne funkcie

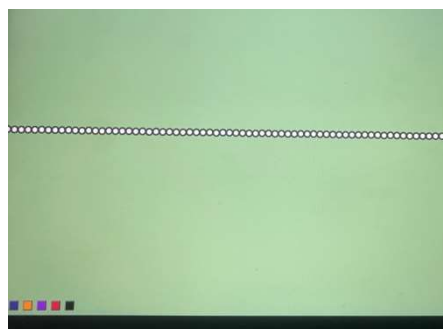
Aby sme zabezpečili funkčné, atraktívne užívateľské rozhranie a uspokojili návštevníkov webu, bolo potrebné vytvoriť niekoľko nastaviteľných atribútov. Užívateľ si tak môže jednoduchú vizualizáciu padajúcich guľôčok ozvláštniť a prispôsobiť jeho aktuálnym potrebám.

2.3.1 Color Picker

Color picker je interaktívny prvok, ktorý nám umožňuje vybrať si farbu z farebnej palety. Prednastavená farba pozadia je zelená, aby sedela k dizajnu našej profesijnej webovej stránky. Ak však klikneme na tlačidlo, zjaví sa menu, kde máme možnosť vybrať kurzorom, kvapkadlom alebo zadať RGB hodnoty, ktoré sú jedinečné pre každú farbu. Následne farbu aplikujeme ako farbu pozadia, tlačidla a výplne guľôčok, keďže majú iba biely obrys.



```
<input type="color" id="color-picker" value="#9bdd9d">
```



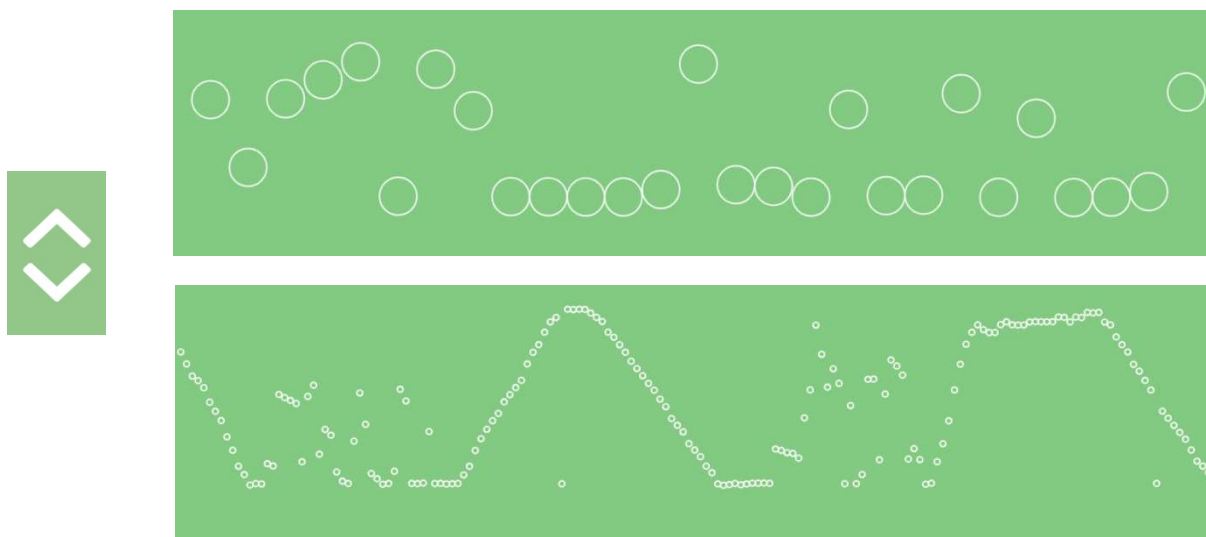
Prvotná verzia programu

Spočiatku sme neplánovali vytvoriť takúto paletku farieb. Na výber bolo iba päť tlačidiel, z ktorých každé predstavovalo jednu farbu. Aj guľôčky mali iný vzhľad, a to bielu výplň a čierny obrys. Po vypočutí rád ostatných maturantov, sme sa rozhodli pozdvihnúť dizajn vizualizéru, čo sa ukázalo ako správne rozhodnutie.

2.3.2 Veľkosť objektov

Keď sme sa pohrali s dizajnom vykresľovaných objektov, rozhodli sme sa pridať ešte dve tlačidlá na zväčšenie a zmenšenie priemeru guľôčiek. Tento proces sa ukázal byť o niečo zložitejší, ako sme si mysleli. Vytvorili sme funkcie v javascripte, ktoré menia priemer guľôčok, ale aj priestor medzi nimi, aby sa nezačali prekrývať či vytvárať obrovské medzery.

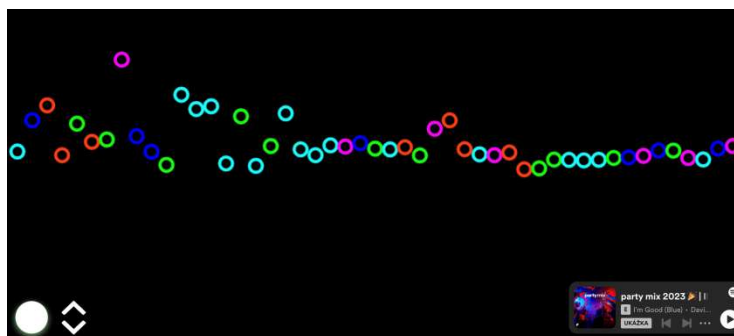
Aby sa zmeny prejavili na stránke po kliknutí na príslušné tlačidlá, využili sme metódu `.getElementById` a `.addEventListener`, ktorá nám umožnila nájsť prvok na základe jeho ID a zmeniť jeho atribúty.



Čím sú guľôčky menšie, tým presnejšie reprezentujú daný zvuk.

2.3.3 Party Mode

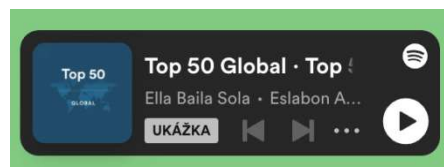
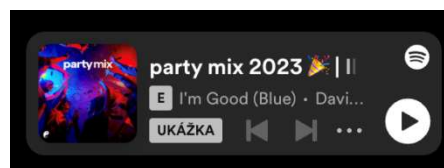
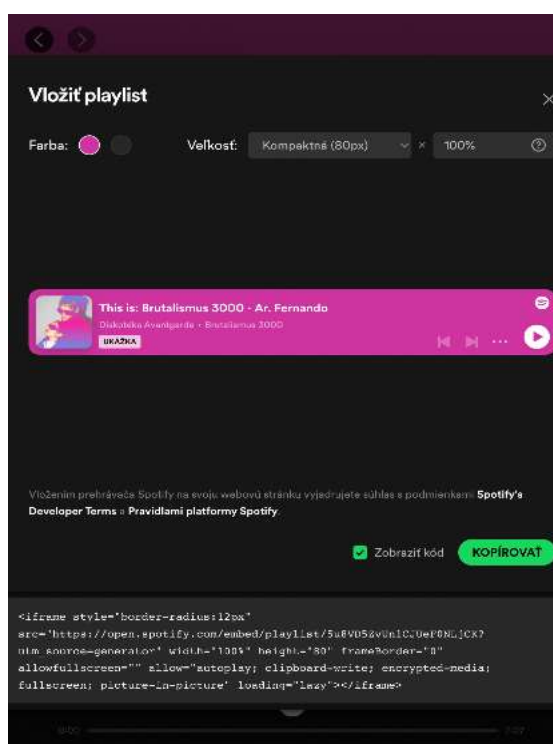
Ďalšie z tlačidiel v ponuke nám umožňuje prekliknúť sa na stránku s prednastaveným „party režimom“. Po kliknutí na čierne tlačidlo sa presunieme hypertextovým odkazom na novú stránku s trvalo čiernym pozadím a pravidelne blikajúcimi farbami guľôčok, ktoré sprevádza Spotify party playlist vhodný pre zábavu. Je zabezpečené pohodlné a intuitívne preklikávanie medzi oboma režimami vizualizérov. Ak sa užívateľ rozhodne vrátiť späť na pôvodnú stránku, môže tak urobiť kliknutím na žiariace biele tlačidlo, ktoré sa nachádza na stránke a funguje na rovnakom princípe ako tlačidlo prekliku na party režim. Týmto spôsobom je zabezpečené pohodlné a intuitívne prekliknutie sa medzi oboma stránkami.



Pre vytvorenie Party Mode sme využili zákonitosti, ktoré platia aj v základnom režime. Stačilo nám pozmeniť iba pár vlastností v animacie.js dokumente - farba pozadia, hrúbka obrysu guľôčok, náhodné striedanie, farieb, playlist...

2.3.4 Spotify playlist

Vizualizéru bola pridaná funkcia prehrávania dvoch pravidelne aktualizovaných albumov od populárnej hudobnej služby Spotify – Top 50 Global a Party mix 2023. Táto funkcia umožňuje užívateľom streamovať hudbu priamo na stránke projektu, bez nutnosti prechádzať na inú webovú aplikáciu. Pomocou jednoduchého kódu možno vytvoriť mini prehrávač ľubovoľného albumu, ktorý zobrazuje meno autora a názov prehrávanej skladby, umožňuje spúšťanie, pozastavenie aj preskakovanie pesničiek. Hudba sa spustí aj používateľom, ktorí nemajú Spotify predplatené, no s určitými obmedzeniami. Ďalej je možné upravovať jeho dizajn a veľkosť pomocou CSS. Prehrávanie hudby tak nie je iba funkčné, no aj vizuálne príjemné.



← Kód na skopírovanie

Záver

Naša ročníková práca potvrdila, že programovanie a umenie sa môžu v mnohých prvkoch prelínať a mať spoločné črty. Tvorba vizualizéru nám umožnila prepojiť naše záľuby v hudbe a informatike a zároveň dokázala, že programovanie môže byť nielen mechanickým, ale aj kreatívnym a zábavným procesom. Naším cieľom bolo vytvoriť aplikáciu, ktorá bude reagovať na aktuálne zvuky okolia a vizuálne doplní hudobný zážitok užívateľa. Výsledný vizualizér bol plne prispôsobiteľný a ponúkal množstvo rôznych nastavení pre užívateľa. Pomocou tejto aplikácie sme dokázali zachytiť zvukový signál a preniesť jeho charakteristické vlastnosti pomocou grafických prvkov na monitor.

V teoretickej časti sme popísali základné poznatky o zvuku a jeho vlastnostiach, ktoré boli nevyhnutné pre pochopenie a vytvorenie vizualizéru. Ozrejmili sme, že zvuk je mechanické vlnenie, ktoré má mnoho aplikácií v bežnom živote a šíri sa prostredím v podobe tlakových vln. Z informatického hľadiska sme sa dozvedeli, že sa zvuk skladá z digitálneho signálu, ktorý reprezentuje zmeny tlaku v čase a pre jeho záznam sa využívajú mikrofón, zvuková karta a softvér.

Celková práca nás naučila veľa a dala nám množstvo nových poznatkov, ktoré v budúcnosti určite využijeme. Pracovali sme s jazykom JavaScript, ktorý bol pre nás novým, no aj napriek tomu sme vedeli nájsť informácie a vynaliezavé riešenia rôznych problémov. Práca ukázala, že programovanie má nielen praktické, ale aj estetické využitie. Výsledný audio vizualizér nie je len užitočným nástrojom na doplnenie hudobného zážitku, ale aj esteticky príjemným dielom, ktorý dokáže prezentovať zvukové vlny ako zaujímavé grafické prvky na obrazovke. Vytváranie vizualizéru nás presvedčilo o tom, že programovanie môže byť skutočné umenie.

Zoznam bibliografických odkazov

CREATIVE CODING INŠPIRÁCIA:

<https://www.youtube.com/watch?v=8zHPGbraW4Q&t=1208s>

ZVUK TEÓRIA:

<https://www.youtube.com/watch?v=SPVi6SDE3H4&t=182s>

https://rayman.gjar-po.sk/moodle/pluginfile.php/11456/mod_resource/content/8/2_ION_z.pdf

AUDIO VIZAULIZÉRY:

https://en.wikipedia.org/wiki/Music_visualization

AKO FUNGUJÚ AUDIO VIZUALIZÉRY:

<https://www.youtube.com/watch?v=G4W1hLLNcic&t=89s>

INŠPIRÁCIA:

<https://www.youtube.com/@Frankslaboratory>

<https://www.youtube.com/watch?v=G-J36pjSthI>

POMOC PRI TVORBE:

<https://www.youtube.com/watch?v=qNEb9of714U>

<https://www.youtube.com/watch?v=VXWvfrmpapI>

SKÁKANIE LOPTIČKY JS:

<https://www.youtube.com/watch?v=F9LUPoH400M>

Zoznam príloh

Príloha A CD s ročníkovou prácou