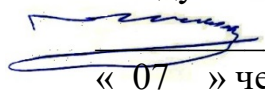


НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
ІМЕНІ ІГОРЯ СІКОРСЬКОГО
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ
КАФЕДРА АКУСТИЧНИХ ТА МУЛЬТИМЕДІЙНИХ ЕЛЕКТРОННИХ СИСТЕМ

«На правах рукопису»
УДК 621.004.522

«До захисту допущено»

Завідувач кафедри

 С. А. Найда
« 07 » червня 2021 р.

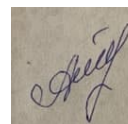
Дипломна робота бакалавра

зі спеціальності (спеціалізації) 171 Електроніка

на тему: «Віртуальний синтезатор звуків для дистанційної освіти»

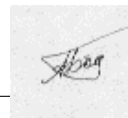
Виконав: студентка IV курсу, групи ДГ-72

Таран Аліна Олександрівна



Керівник доцент каф. АМЕС к.т.н. доц. Богданов О. В.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)



(підпис)

Консультант _____

(назва розділу) (посада, науковий ступінь, вчене звання, прізвище, ініціали)

(підпис)

Рецензент доцент каф. КЕОА ФЕЛ к.т.н. доц. Кучернюк П. В.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

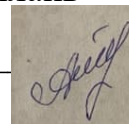
(підпис)

Засвідчую, що у цій дипломній роботі

немає запозичень з праць інших

авторів без відповідних посилань

Студент



Київ – 2021 р.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет Електроніки

Кафедра Акустичних та мультимедійних електронних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 171 «Електроніка»

ЗАТВЕРДЖУЮ

Завідувач кафедри

 С. А. Найда

«_ 07 _» _червня_ 2021 р.

ЗАВДАННЯ

на дипломну роботу студенту

Таран Аліні Олександрівні

(прізвище, ім'я, по батькові)

керівник проекту (роботи) Богданов Олексій Вікторович, доц. к.т.н.
затверджені наказом по університету від «24» травня 2021 р. № 1316-с

2. Строк подання студентом роботи 01 червня 2021 року

3. Вихідні дані до роботи.

Існуючі типи синтезу звуку, його фізичні принципи, створення веб-сайту віртуального синтезатора.

4. Зміст дипломної роботи (перелік завдань, які потрібно розробити): розглянути існуючі пристрої та програми синтезу звуку; обрати метод формування звуку на веб сторінці; реалізувати програмний модуль.
5. Перелік графічного (ілюстративного) матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо) презентація PowerPoint
6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

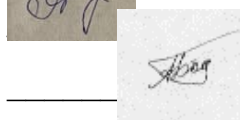
7. Дата видачі завдання 12 квітня 2021 року

Календарний план

№ з/п	Назва етапів виконання дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Аналіз сучасних методів синтезу звуку	12.04 – 18.04	Виконано
2	Написання першого розділу	19.04 – 27.04	Виконано
3	Написання другого розділу	28.04 – 09.05	Виконано
4	Оформлення звіту та щоденника	10.05 – 16.05	Виконано
5	Написання третього розділу	17.05 – 31.05	Виконано
6	Написання реферату та оформлення всієї роботи	01.06 – 03.06	Виконано

Студент

Керівник проекту (роботи)

А. О. Таран

О. В. Богданов

РЕФЕРАТ

Таран, А. О. *Віртуальний синтезатор звуків для дистанційної освіти* : дипломна робота бакалавра : 171 Електроніка. – Київ, 2021. – 41 с.

Ключові слова: веб-сервіси, інтерфейси прикладного програмування, адаптивне навчання, акустичні програми.

Проаналізувавши ситуацію в сучасному світі, можна зробити висновок, що в будь-який момент може відбутись ситуація, коли немає фізичної змоги виходити з дому. З цієї причини метою даної розробки було створення віртуального аналогового синтезатора, який би містив стандартні налаштування музичного інструменту.

В ході виконання даної роботи було розглянуто основні типи синтезу звуків, порівняння реального та програмного синтезатора. Для розробленого сайту використано операторний синтез, так як це стандарт у прикладному програмному інтерфейсі, який застосовується на сайті.

За допомогою обраного прикладного програмного інтерфейсу Web Audio API, розроблено веб-застосунок віртуального синтезатору. Це дасть можливість відтворювати звук справжнього синтезатора з такими налаштуваннями, як:

- вибір фільтру;
- вибір форми хвилі;
- регулювання звуку;
- регулювання частоти.

ABSTRACT

Keywords: web services, application programming interfaces, adaptive learning, acoustic applications.

Analyzing the situation in the modern world, we can conclude that at any time there may be a situation where there is no physical ability to leave home. For this reason, the purpose of this work was to create a virtual analog synthesizer that would contain the standard settings of a musical instrument.

In the course of this work the main types of sound synthesis, comparison of real and software synthesizer were considered. Frequency Modulation(FM) synthesis is used for the site developing, as it is a standard in the application programming interface used on the site.

Using the Web Audio API a virtual synthesizer web application was developed. This will allow you to play the sound of a real synthesizer with the following settings:

- filter selection;
- waveform selection;
- sound control;
- frequency control.

Зміст

Реферат 4

ABSTRACT.....	5
----------------------	----------

ВСТУП 8

1. РОЗДІЛ 1. Синтез звуків.....	9
1.1. Актуальність	9
1.2. Історія	9
1.3. Основні типи синтезу звуку	11
1.3.1. Підсумовувальний (адитивний) синтез	12
1.3.2. Субтрактивний синтез	12
1.3.3. Операторний(FM, від англ. Frequency Modulation)	12
1.3.4. Семплерний(sample-вибірка)	13
1.3.5. Табличний(англ. Wavetable).....	14
1.3.6. Фізичний синтез(модулювання)	14
1.3.7. Технологія WaveGuide.....	14
1.4. Порівняння апаратного та програмного синтезатора	15
Висновок до розділу 1	16
2. РОЗДІЛ 2.....	17
2.1. Фізичні принципи синтезу звуку	17
2.1.1. Звукові хвилі та осцилятори.....	17
2.1.2. Основні форми хвиль	18
2.1.3. Фільтри	20
2.1.4. Властивості фільтрів.....	21
2.1.5. ADSR-обвідна(англ. ADSR-envelope).....	21
2.2. Класифікація синтезаторів	22
2.2.1. Різниця між аналоговим та цифровим звуком.	22
Висновок до розділу 2	24
3. РОЗДІЛ 3. Створення веб-сайту.....	25
3.1. Використані технології та вибір мови програмування	25
3.2. Web Audio API [1].....	25
3.2.1. Порядок дій виконання маніпуляцій над аудіо	25

	7
3.2.2. Інтерфейси.....	26
3.3. Створення веб-сайту.	29
3.3.1. Вибір середовища для програмування.	29
3.3.2. Додавання React.....	29
3.3.3. Створення компонентів.	31
3.3.4. Створення синтезатора.....	34
3.4. Запуск програми.....	37
Висновки	38
Перелік посилань.....	39
Додаток 1. Лістинг програми	1

ВСТУП

На даний момент велика кількість музикантів, які складали електронну музику, використовують аналогові синтезатори (в основному їх віртуальні форми). Вони віддають перевагу саме цьому типу синтезаторів, оскільки йому притаманні унікальне звучання і гнучка система налаштування звуку. Так, як подібні комерційні програмні продукти мають високу вартість і не завжди доступні користувачеві. Також, проаналізувавши ситуацію в сучасному світі, можна зробити висновок, що в будь-який момент може відбутись ситуація, коли немає фізичної змоги виходити з дому. З цієї причини метою даної розробки було створення віртуального аналогового синтезатора, який би містив стандартні налаштування музичного інструменту.

1. Синтез звуків

1.1. Актуальність

В даний час індустрія виготовлення звукового контенту нараховує багато пристроїв синтезу звуку, серед них є і синтезатор. Існує безліч синтезаторів, які відрізняються різними налаштуваннями: якістю звуку, об'ємами, тощо. Це все додає до вартості кругленьку суму. Тому що, ці пристрої у сфері звукового синтезу можна назвати «зразковими» по звучанню. Також, проаналізувавши ситуацію в сучасному світі, можна зробити висновок, що влюбий момент, може відбутись ситуація, коли немає фізичної змоги виходити з дому. Звідси постає проблема в освітніх програмах, а саме потрібна можливість дистанційного навчання. Тому, дана робота є актуальною, як мінімум, з двох причин: вирішення проблеми дорогої вартості пристроїв та спосіб дистанційного навчання. Як аналог реального синтезатора, було розроблено програмний.

Програмний синтезатор — це віртуальний музичний інструмент, який синтезує звук за допомогою комп'ютера. Синтезатор дає можливість створити будь-який звук, який тільки можна уявити.

1.2. Історія

Як вказано в джерелі [2], першим комп'ютером, який був використаний для створення музики, вважається CSIRAC (аббр. від англ. Council for Scientific and Industrial Research Automatic Computer) — перша австралійська цифрова ЕОМ і четверта в світі ЕОМ з програмною пам'яттю.

CSIRAC спочатку називався CSIR Mk 1, і в даний час є єдиним в світі уцілілим комп'ютером першого покоління. Він був спроектований в кінці 1940-х років групою австралійських інженерів під керівництвом Тревора Пірсі і Мастона Берда, які працювали незалежно від своїх колег в Європі і США.

Перша тестова програма CSIRAC була апробована в кінці 1949 року, а в період з 1950 до 1951 року машину використовували в цілях створення музики, для чого математик Джеф Хілл розробив спеціальну алгоритмічну програму для

комп'ютерного синтезу машинного звучання популярних музичних творів того часу.

У 1951-му році CSIRAC успішно «виконав» популярний шлягер «Марш полковника Боги» (англ. «Colonel Bogey March»), проте в подальших дослідженнях і розробках по створенню музики цей комп'ютер не був задіяний.

У 1951 році у Великобританії фахівцями виїзної студії BBC в Манчестерській лабораторії обчислювальних машин був зроблений перший запис музики, згенерованої комп'ютером. Машина, створена Тюрінгом і займала майже весь перший поверх лабораторії, могла генерувати три мелодії — «Боже, бережи Королеву» (англ. God Save the King), «Бе Бе Чорна овечка» (англ. Baa, Baa Black Sheep) і класику свінгу(особливий джазовий стиль музики) «У настрої» (англ. In the Mood) Глена Міллера. Музика записана на 12-дюймовий (30.5 см) ацетатний диск. При цьому фундаментальні роботи Тюрінга кінця 1940-х років по перетворенню комп'ютера в музичний інструмент виявилися непоміченими. Звуковий артефакт, який представляє Тюрінга як музичного новатора, відновлений в 2016 році.

Відомо з джерела [3], що з появою Commodore 64 комп'ютерна музика виходить на новий рівень. У комп'ютері був встановлений звуковий чіп SID (Sound Interface Device), який дозволив користувачам створювати музику. Слід зазначити, що звук, якого дозволяв домогтися звуковий чіп SID, популярний і до цього дня — для Mac і PC доступні емулятори.

Створення MIDI (Musical Instrument Digital Interface) має неоціненне значення для всієї музичної індустрії. Цей новий протокол дозволяв з'єднати комп'ютер з іншим обладнанням. Якби не MIDI, історія розвитку комп'ютерних технологій в музиці була б зовсім іншою. В середині 1980-х створено Atari ST. Комп'ютер був оснащений MIDI-інтерфейсом, що робило його ідеальною платформою для всіх, хто хотів приєднати зовнішнє обладнання до свого комп'ютера.

У 1989 році компанія Steinberg випускає програмне забезпечення Cubase. У першій версії робочої станції DAW (аббр. від англ. Digital Audio Workstation) було

введено інтерфейс, який став стандартом для всіх сучасних секвенсорів — вертикальний список треків і горизонтальний таймлайн (Рис. 1.1).

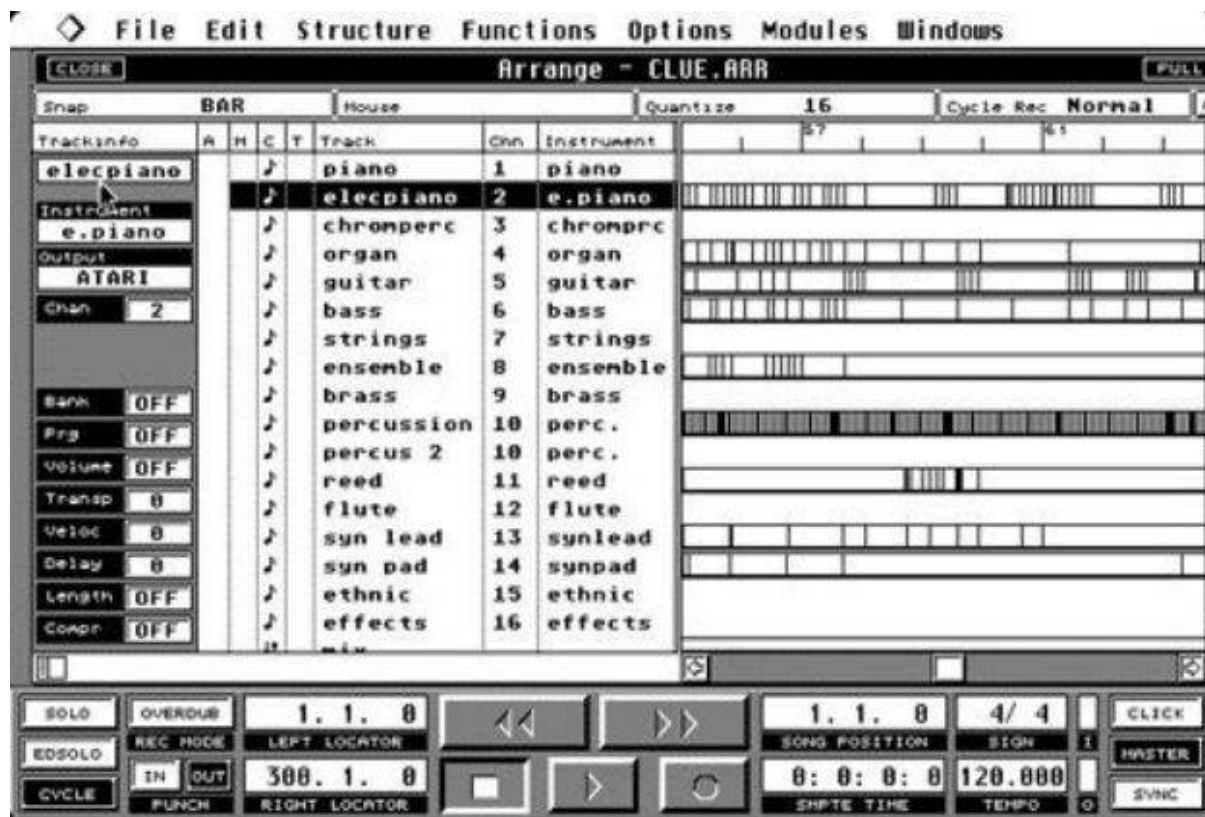


Рис. 1.1 Робоче вікно Cubase 1989 року

У 1997 році випускається Cubase VST (аббр. від англ. Virtual Studio Technology). Появилась можливість не тільки записувати декілька звукових доріжок на комп'ютері, а також обробляти їх за допомогою різних ефектів.

У 1999 році появляються перші повноцінні VST-плагіни і VST-інструменти. Прихильність Steinberg до інновацій була продемонстрована ще раз, коли був випущений Cubase VST 3.7. З'явилася можливість підключення програмних інструментів безпосередньо в DAW. Але ж сьогодні ми вже звикли приймати це як належне.

1.3. Основні типи синтезу звуку

Синтезатори використовують різні типи синтезу звуку, серед яких слід зазначити такі:

1.3.1. Підсумовувальний (адитивний) синтез

Як вказано в джерелі [4] синтез заснований на твердженні Фур'є про те, що будь-яке періодичне коливання можна представити у вигляді суми чистих тонів (синусоїдальних коливань з різними частотами і амплітудами).

Для цього потрібен набір з декількох синусоїдальних генераторів з незалежним управлінням, вихідні сигнали яких підсумовуються для отримання результуючого сигналу. На цьому методі ґрунтується принцип створення звуку в духовому органі.

Переваги методу: дозволяє отримати будь-який періодичний звук, і процес синтезу добре передбачуваний (зміна настройки одного з генераторів не впливає на решту спектру звуку).

Основний недолік: для звуків складної структури можуть знадобитися сотні генераторів, що досить складно і дорого реалізувати. Для зниження вартості реалізації замість набору окремих генераторів (реальних або математичних) застосовується зворотне перетворення Фур'є.

1.3.2. Субтрактивний синтез

З джерела [4] відомо, що цей метод заснований на відніманні елементів. У синтезаторах визначальним елементом субтрактивного синтезу є наявність фільтрів. Фільтр «вирізає» частину спектра зі звуку, формуючи потрібне тембральне забарвлення.

Переваги методу: відносно проста реалізація і досить широкий діапазон синтезованих звуків. На цьому методі побудовано безліч студійних і концертних синтезаторів.

Недолік: для синтезу звуків зі складним спектром потрібна велика кількість керованих фільтрів, які досить складні і дорогі.

1.3.3. Операторний (FM, від англ. Frequency Modulation)

Як вказано в джерелі [4] спосіб синтезу звукових хвиль, в основі якого лежить частотна модуляція коливання простої форми (наприклад, синусоїдальної). Результатом є звукова хвиля більш складної форми, що відрізняється по звучанню від вихідного коливання.

Переваги методу: завдяки простоті цифрової реалізації, метод набув широкого поширення в студійній і концертній практиці. Також метод дає широкі можливості по синтезу різного роду ударних звуків, та різних звукових ефектів, недосяжних в інших методах.

Недоліки: практичне використання цього методу досить складне через те, що більша частина звуків, одержуваних з його допомогою, є шумоподібні коливання, і досить лише злегка змінити налаштування одного з генераторів, щоб чистий тембр перетворився в шум.

1.3.4. Семплерний (sample-вибірка)

Як вказано в джерелі [4], у цьому методі записується реальне звучання (семпл), яке потім в потрібний момент відтворюється. Для отримання звуків різної висоти відтворення прискорюється або сповільнюється. При незмінній швидкості вибірки застосовується розрахунок проміжних значень відліків (інтерполяція). Щоб тембр звуку при зсуві висоти не змінювався надто сильно, використовується кілька записів звучання через певні інтервали (зазвичай — через 1-2 октави). У ранніх семплерних синтезаторах звуки в буквальному сенсі записувалися на магнітофон, в сучасних застосовується цифровий запис звуку.

Для зменшення необхідного обсягу пам'яті застосовується зациклення семпла. В цьому випадку записується тільки короткий час звучання інструменту, потім в ньому виділяється середня фаза зі сталим звуком, яка при відтворенні повторюється до тих пір, поки включена нота (натиснута клавіша), а після відпускання відтворюється кінцева фаза.

Насправді, цей метод не можна з повним правом називати синтезом — це скоріше метод запису-відтворення. Однак в сучасних синтезаторах на його основі відтворений звук можна піддавати різній обробці — модуляції, фільтрування, додаванню нових гармонік, звукових ефектів, в результаті чого звук може набувати зовсім новий тембр, іноді зовсім не схожий на початковий.

Переваги методу: дозволяє отримати точну подобу звучання реального інструмента, однак для цього потрібні чималі обсяги пам'яті.

Недоліки: запис звучить природно лише при тих самих параметрах, при яких вона була зроблена при спробі.

1.3.5. Табличний (англ. Wavetable)

Як вказано в джерелі [4], що це різновид семплерного методу, коли записується не всі звучання цілком, а його окремі фази — атака, початкове загасання, середня фаза і кінцеве згасання, що дозволяє різко знизити обсяг пам'яті, необхідний для зберігання семплів.

Ці фази записуються на різних частотах і при різних умовах. В результаті чого виходить сімейство звучань одного інструмента. При відтворенні ці фази потрібним чином складаються.

Переваги методу: можливість при відносно невеликому обсязі семплів отримати досить широкий спектр різних звучань інструмента, а головне — помітно посилити виразність звучання.

Недоліки: складність сполучення різних фаз один з одним, щоб переходи не сприймали на слух, і звучання було цілним і безперервним.

1.3.6. Фізичний синтез (модулювання)

Як вказано в джерелі [4], що синтез складається в моделюванні фізичних процесів, що визначають звучання реального інструмента на основі його заданих параметрів (наприклад, для скрипки — порода дерева, склад лаку, геометричні розміри, матеріал струн і смичка і т. д.). У зв'язку з крайньою складністю точного моделювання навіть простих інструментів і величезним обсягом обчислень метод поки розвивається повільно, на рівні студійних і експериментальних зразків синтезаторів. Очікується, що з моменту свого достатнього розвитку він замінить відомі методи синтезу звучань акустичних інструментів, залишивши їм тільки завдання синтезу, які не зустрічаються в природі тембрів.

1.3.7. Технологія WaveGuide

Як вказано в джерелі [4], ця технологія активно розробляється в Стенфордському Університеті і застосовується вже в декількох промислових моделях електронних роялів. Являє собою різновидність фізичного моделювання, при якій моделюється поширення коливань, представлених дискретними

відліками по струні (одномірне моделювання) і по резонансним поверхням (двовимірне моделювання) або в об'ємному резонаторі (тривимірне). При цьому з'являється можливість моделювати також нелінійні ефекти, наприклад удар молотка, а також взаємний зв'язок струн і зв'язок горизонтальної і вертикальної мод.

1.4. Порівняння апаратного та програмного синтезатора

Апаратні та програмні синтезатори мають свої плюси і мінуси. З одної сторони більше переваг буде у реального синтезатора, з другої у програмного, але загальна тенденція така: реальні синтезатори будуть завжди дорожчими, але якіснішими по звучанню, а комп'ютерні більш гнучкими. Можна порівнювати їх по таких категоріях:

- Доступність

Як уже було зазначено, реальний синтезатор буде набагато дорожчий, отже доступність в нього менша. Базова стандартна модель синтезатора, для початківця буде вартувати починаючи від 1800 грн, на момент написання даної роботи. Інформацію взято з джерела [5].

Щодо розробленого сайту, він є доступний для будь-якого користувача.

- Ефекти та управління

На даний момент, у реального синтезатора буде більше налаштувань. Але, потрібно зазначити, що програмний можна неперервно розвивати, так як технологія, яку використано на розробленому сайті також розвивається і періодично можна добавляти новий функціонал.

- Габарити та вага

Взято габарити та вагу синтезатора Casio SA-47.

Вага: 1кг

Габарити: 446 x 208 x 51 mm

Інформацію взято з джерела [6].

Віртуальний синтезатор можна запустити на різних пристроях(телефон, планшет, комп'ютер), тому вага і габарити будуть різні.

Висновок до розділу 1

Перший розділ був присвячений огляду задачі синтезу звуків. В цьому розділі розглянуто основні типи синтезу звуків, актуальність даної теми, порівняння реального та програмного синтезатора. Для розробленого сайту було використано операторний синтез, так як це стандарт у прикладному програмному інтерфейсі, який застосовується на сайті.

2. РОЗДІЛ 2

2.1. Фізичні принципи синтезу звуку

Є ряд пристроїв, які створюють і модифікують сигнали, що використовуються для синтезу звуку. Далі буде розглянуто кожен з них.

2.1.1. Звукові хвилі та осцилятори

Як вказано в джерелі [7], звуки — це хвилі тиску, які рухаються по повітрю або іншому середовищі до наших вух. На відміну від хвиль в океані, які рухаються вгору і вниз, хвилі тиску рухаються вперед і назад. Ці хвилі рухають наші барабанні перетинки всередину і назовні, і ми сприймаємо це як звук. Синтез звуку — це «мистецтво» створення сигналів, які, перетворюючись динаміком на звукові хвилі, людям здаються цікавими. Є ряд пристроїв, які створюють і модифікують сигнали, що використовуються для синтезу звуку.

Перший такий пристрій, який ми розглянемо, називається *осцилятором*. Він створює послідовний повторюваний сигнал. Сигнали від осциляторів та інших джерел використовуються для управління рухом дифузоров у наших динаміках, які створюють справжні звукові хвилі, що рухаються до наших вух. Якщо ви прив'язали один кінець мотузки до дверної ручки, відступили кілька футів назад і дуже швидко поворушили другим кінцем мотузки вгору-вниз, ви робите приблизно те саме, що і осцилятор. Різниця полягає в тому, що ви хитаєтеся мотузкою, тоді як генератор хитає звуковим сигналом.

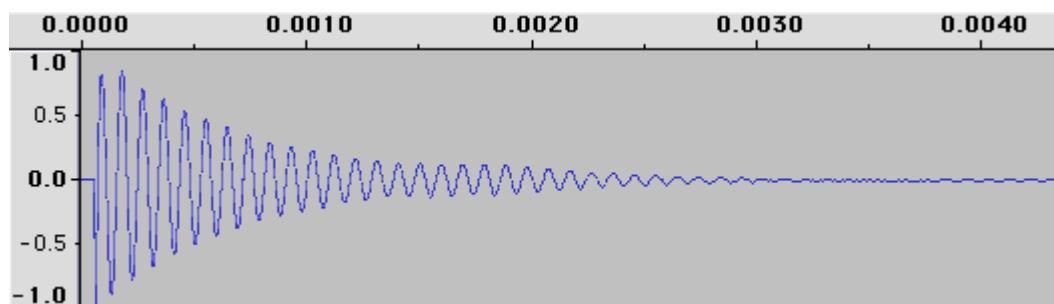


Рис. 2.1 Графік звукового сигналу

Звукові сигнали часто представлені на графіку, де горизонтальна вісь x представляє *час*, а вертикальна вісь y — *тиск сигналу* (Рис. 2.1). Це називається поданням звуку в часовій області. Графіки часової області — це щось на зразок

інструкцій для спікерів про те, як рухатись та виходити. Коли на графіку показується 1, дифузор динаміка повністю виштовхується, коли показується -1, дифузор втягується всередину. Цей рух створює хвилю тиску в повітрі, яку ми чуємо як звук. Якщо дифузор динаміка рухається всередину та назовні відповідно до графіку вище, він видаватиме звук барабану.

2.1.2. Основні форми хвиль

З джерела [7] відомо, що існують чотири основні форми хвиль:

- Синусоїдальна

Синусоїда є найбільш базовою, чистою формою хвилі, з контуром, що описується тригонометричною функцією, яка називається синусоїдою. Будь-яка інша форма сигналу, включаючи інші прості форми сигналів, описані нижче, може бути створена додаванням синусоїд.

Ці хвилі схожі на «м'яку» хвилю в мисці з водою, рухаючись вгору-вниз без різких стартів і зупинок. Загальні звуки, подібні до синусоїди, включають свист, повітря, що дме через отвір порожньої пляшки, і дзвінкий камертон.

- Прямокутна

Амплітуда даної гармоніки в прямокутній хвилі дорівнює оберненому до її гармонічного числа. Наприклад, квадратна хвиля з основною частотою 100 Гц мала б гармоніку на 300 Гц ($100 * 3$) з амплітудою 0,33 ($1/3$).

Прямокутні хвилі мають насичений звук, який не такий густий, як пилкоподібні хвилі, але не такий чистий, як синус. Як і пилкоподібні хвилі, прямокутні хвилі можуть генеруватися додаванням серії синусоїд із зменшенням гучності. Однак прямокутна містить лише непарні гармоніки.

- Трикутна

Амплітуда даної гармоніки у трикутній хвилі дорівнює оберненому квадрату її гармонічного числа. Наприклад, хвиля трикутника з основною частотою 100 Гц і амплітудою 1 мала б гармоніку на 300 Гц ($100 * 3$) з амплітудою 0,1111 ($1/3^2$).

Трикутні хвилі звучать як щось середнє між синусоїдою та прямокутною хвилею. Як і прямокутна хвиля, вони містять лише непарні гармоніки основної частоти. Вони відрізняються від прямокутних тим, що гучність кожної доданої гармоніки падає швидше.

- Пилкоподібна

Амплітуда даної гармоніки в пилозубій хвилі дорівнює оберненому до її гармонічного числа. Наприклад, пилозуба хвиля з основною частотою 100 Гц та амплітудою 1 мала б гармоніку на 200 Гц ($100 * 2$) з амплітудою 0,5 ($1/2$), інша гармоніка на 300 Гц ($100 * 3$) з амплітуда 0,33 ($1/3$) тощо. Чим більше гармонік додано таким чином, тим більше хвиля буде виглядати як ідеалізована пилоподібна хвиля, зображена на рисунку.

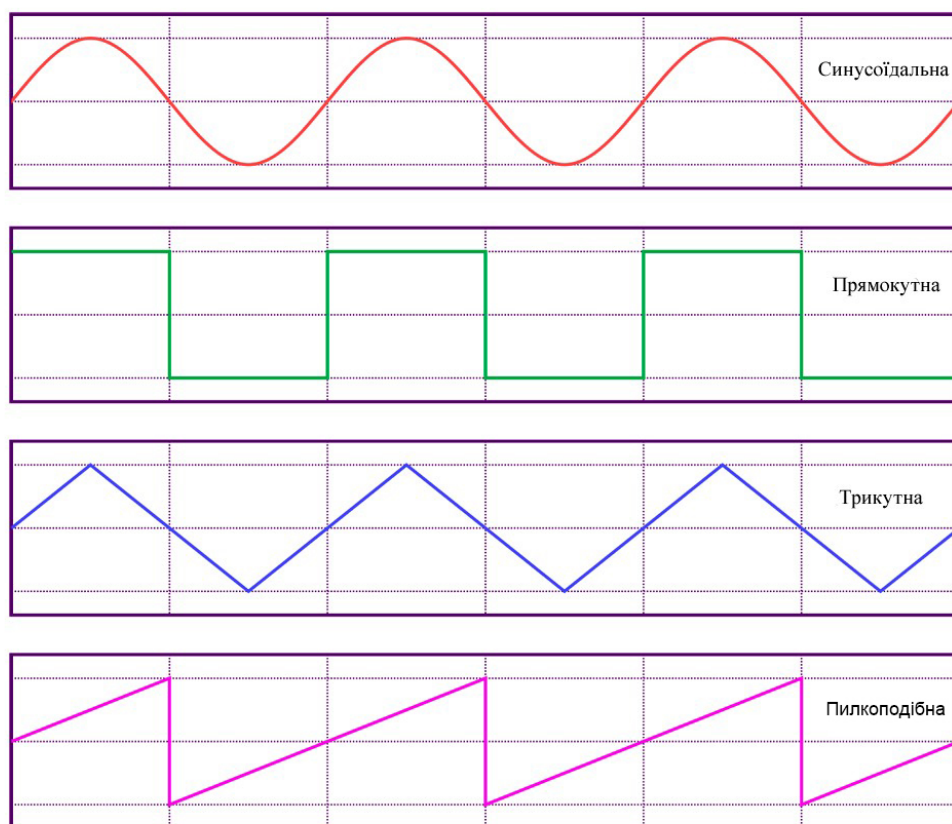


Рис. 2.2 Форми хвиль [8]

Пилкоподібні хвилі, мають дуже сильний, чистий, гудкий звук. Її можна зробити, додавши ряд синусоїд на різних частотах і рівнях гучності. Частота першої, найгучнішої синусоїди — це те, що ми чуємо як частоту результуючого

пилкоподібного зубця. Це називається основною частотою. Кожна з інших, поступово тихіших синусоїдальних хвиль, що складають пилкоподібний зуб, має частоти, які є цілими кратними основній частоті. Ці частоти називаються гармоніками.

2.1.3. Фільтри

З джерела [7] відомо, що фільтри — це інструменти для маніпулювання сигналами. Будь-який пристрій, який будь-яким чином модифікує сигнал, технічно є фільтром. Однак, коли люди говорять про фільтри, вони зазвичай мають на увазі фільтри, які змінюють гармонічний вміст сигналу, змінюючи характеристики звуку в частотній області.

Фільтри дозволяють вибрати діапазон частот у звуці, або посилити або зменшити ці частоти. Зменшення високих частот або збільшення низьких частот усередині звуку робить його «темнішим» або приглушеним, тоді як збільшення високих частот або зменшення низьких частот робить звук «яскравішим». Таке фільтрування відбувається в реальному житті постійно. Якщо ви розмовляєте з кимось і тримаєте перед ротом велику книгу, книга відфільтровує значну частину високочастотного вмісту вашого голосу, змушуючи його звучати «темно» і приглушено.

- Фільтри низьких та високих частот

Фільтр низьких частот дозволяє низьким частотам проходити крізь фільтр і блокує високі частоти, в результаті чого — звук приглушений. Діапазон частот, заблокованих фільтром, називається діапазоном зупинки. Діапазон частот, які проходять через фільтр, називається смуга пропускання. Перехід від смуги пропускання до смуги зупинки відбувається поступово і відбувається в діапазоні, який називається смуга переходів. Ширина смуги переходів залежить від швидкості, з якою фільтр зменшує сигнал. Ця швидкість називається крутизною(нахилом), який вимірюється в децибелах на октаву. Частота, коли фільтр знижує рівень сигналу приблизно до 0.7 від його початкового рівня

називається частотою відсічення. Фільтр високих частот робить протилежність фільтру низьких частот: блокує низькі частоти і пропускає високі частоти.

- **Смугові фільтри**

Смуговий фільтр — це як фільтр низьких частот і фільтр високих частот, що використовується в поєднанні для виділення групи частот, через які проходять, поки все інше вирізається.

2.1.4. Властивості фільтрів

Відомо з джерела [7], що фільтри мають ряд атрибутів, над якими ми маємо певний контроль. Для фільтрів низьких і високих частот ми можемо змінити частоту відсічення, дозволяючи контролювати діапазон частот, на які впливаємо. Проведення синтезатора через фільтр низьких частот та повільне змінення частоти відсічення від високої до низької та назад — популярний прийом, який використовується в електронній танцювальній музиці. За допомогою смугових фільтрів, замість того, щоб змінювати частоту відсічення, ми змінюємо центральну частоту та ширину зрушеної смуги. Також за допомогою деяких фільтрів ми можемо змінювати нахил, який визначає, наскільки швидко зменшуються частоти смуги зупинки.

Багато фільтрів дозволяють змінювати резонанс. Він виникає, коли звук в смузі пропускання поблизу відсічної частоти надходить назад у фільтр, коли він виходить, створюючи зворотний зв'язок. Величина зворотного зв'язку впливає на гучність цих частот, а також на тембр звуку. Ефект wah-wah — це резонансний фільтр низьких частот, що контролює частоту відсічення.

2.1.5. ADSR-обвідна (англ. ADSR-envelope)

В джерелі [9] вказано що, якщо натиснути клавішу синтезатора та відпустити її, звук поступово змінюється з часом. По-перше, він швидко «атакує» (англ. attack) до максимального обсягу. Далі звук «спадає» (англ. decay) на нижчий рівень і залишається там, або «утримується» (англ. sustain). Нарешті, коли ми відпускаємо клавішу, звук «відновлюється» (англ. release) і стихає до тиші. Але, потрібно також розуміти, що у кожного інструмента звукові хвилі

змінюються з різним плином часу. Наприклад, у фортепіано фаза атаки досить повільна, і фаза утримування може бути дійсно довгою, оскільки ноти забирають багато часу. Але на флейті фаза атаки швидша і гостріша, спад майже не відбувається, утримання триває до тих пір, поки флейтист продовжує дути, і відновлення також відбувається дуже швидко.

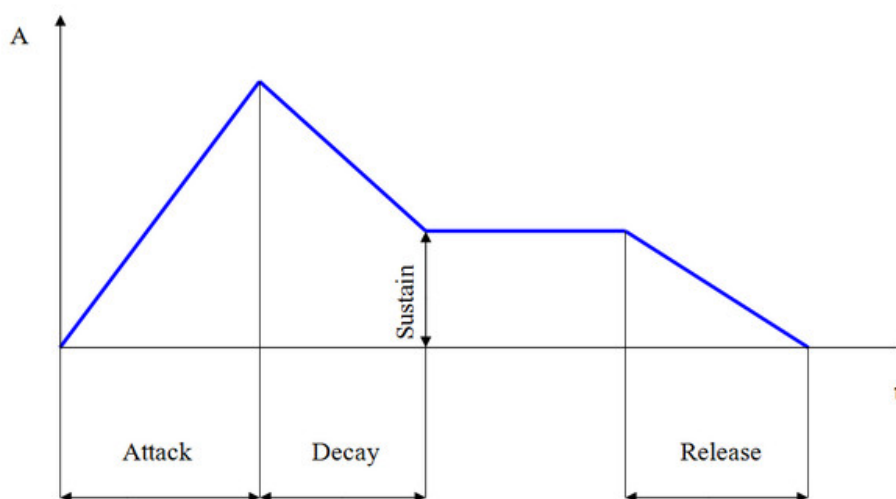


Рис. 2.3 ADSR-обвідна [9]

2.2. Класифікація синтезаторів

Ще в середині двадцятого століття з'явився перший електронний синтезатор — музичний інструмент, здатний створювати звук за допомогою різних способів синтезу. На сьогоднішній день існує декілька технологій виробництва цього інструменту, в залежності від яких визначається тип музичного синтезатора. Відомо з джерела [10], що всього налічується чотири типи синтезатора: аналоговий, цифровий, цифровий з аналоговим синтезом і цифровий з віртуально-аналоговим синтезом. Основною різницею є синтез звуку.

2.2.1. Різниця між аналоговим та цифровим звуком.

Аналоговий відноситься до звуку, записаного за допомогою методів, що відтворюють вихідні звукові хвилі. Вінілові платівки та касети є прикладами аналогових носіїв інформації. Цифровий звук — це спосіб збереження звуку, при якому аудіосигнали перетворюються в серію імпульсів, які відповідають шаблонам двійкових цифр (тобто 0 і 1) і записуються на поверхні магнітної

стрічки або оптичного диска, як вказано в джерелі [11]. Компакт-диски та файли Мр3 є прикладами цифрових носіїв.

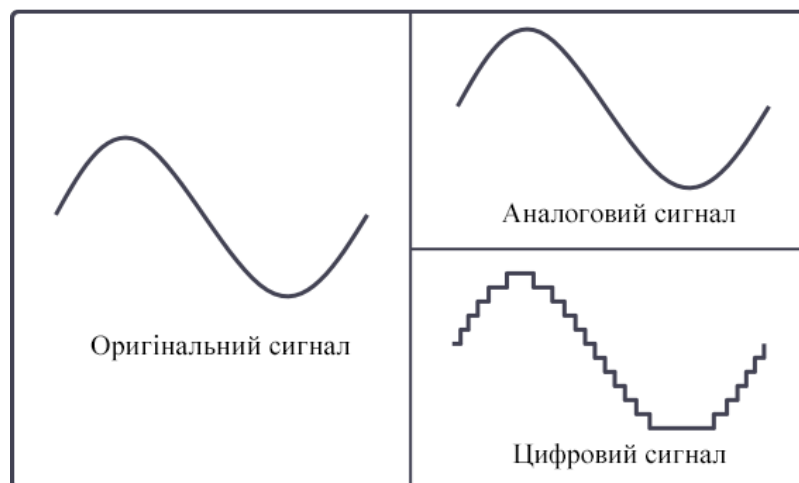


Рис. 2.4 Різниця між аналоговим та цифровим сигналом [12]

Як видно з діаграми на Рис. 2.4, аналоговий сигнал копіює вихідний сигнал, тоді як цифровий сигнал копіює лише вибірккові ділянки вихідного сигналу. Потенційна вірність аналогового запису залежить від чутливості обладнання та носія, що використовується для запису та його відтворення. Серед інших факторів, точність цифрового звуку в значній мірі залежить від швидкості, з якою записуюча апаратура здійснювала вибірку вихідного сигналу протягом певного періоду часу. Навіть використовуючи новітні технології та техніки, цифровий звук все одно не може створити точної копії оригінальної звукової хвилі, як вказано в джерелі [12].

Висновок до розділу 2

У другому розділі описано фізичний принцип синтезу звуку. Також був детальний опис основних форм хвиль, фільтрів та їх властивостей. Була описана класифікація синтезаторів. Так як налічується 4 типи синтезаторів такі, як: аналоговий, цифровий, цифровий з аналоговим синтезом і цифровий з віртуально-аналоговим синтезом, було описано різницю між аналоговим та цифровим звуком. Для більш широко доступу учнів до основ із синтезу звуків, було прийнято рішення створити он-лайн платформу.

3. Створення веб-сайту

3.1. Використані технології та вибір мови програмування

Для розробки сайту було використано мову програмування JS, а конкретніше, бібліотеку ReactJS.

JS — це динамічна мова програмування, реалізовує стандарт ECMAScript. Вона найкраще підходить для цієї роботи, тому що найчастіше використання JavaScript — це створення сценаріїв веб-сторінок.

Також, в даній роботі було використано Web Audio API.

Як вказано в [1], Web Audio API — потужний та багатогранний інструмент для маніпуляції звукової складової на веб-сторінці, що дає можливість розробникам вибрати джерела, додати до них спеціальні звукові ефекти, візуалізувати їх і багато іншого.

Згідно до джерела [13], API — це набір чітко визначених методів для взаємодії різних компонентів. API надає розробнику засоби для швидкої розробки програмного забезпечення. API може бути для веб-базованих систем, операційних систем, баз даних, апаратного забезпечення, програмних бібліотек.

3.2. Web Audio API

Як вказано в джерелі [1]. Це інструмент, який дозволяє обробляти операції над аудіо за допомогою спеціального аудіо контексту (audio context), і був спроектований на використання модульної маршрутизації (modular routing). Базові операції виконуються за допомогою аудіо вузлів (audio nodes), які об'єднуються разом, формуючи таблицю аудіо-маршрутизатора (audio routing graph). Кілька джерел — з різними видами поточних схем — підтримуються навіть зсередини простого контексту. Ця модульна концепція забезпечує гнучкість в створенні складних функцій для динамічних ефектів.

3.2.1. Порядок дій виконання маніпуляцій над аудіо

Відповідно до джерела [1], типовий порядок дій виконання маніпуляцій над аудіо виглядає так:

1. Створимо аудіо контекст.

2. Всередині створеного контексту визначимо джерела — такі як <audio>, генератор (oscillator), потік.
3. Визначимо вузли ефектів, такі як реверберація (reverb), біквадратних фільтр (biquad filter), панорамування (panner), стиснення (compressor).
4. Виберемо кінцеву точку аудіо сигналу, наприклад ваші системні звукові пристрої.
5. Прив'яжемо наші джерела до ефектів, і ефекти до кінцевого сигналу.

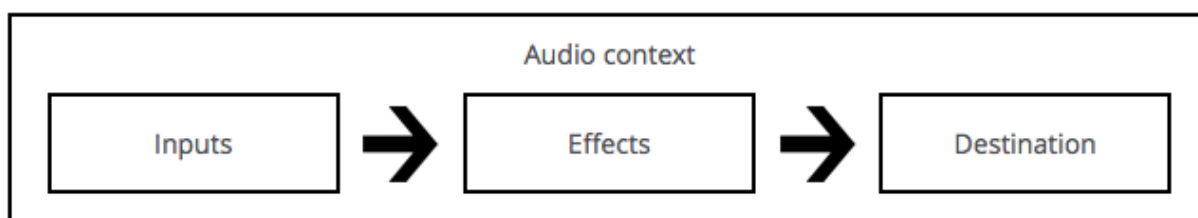


Рис. 3.1 Схематичне представлення порядку дій виконання маніпуляцій над аудіо [1]

3.2.2. Інтерфейси

Всі терміни використані відповідно до [1].

У Web Audio API є всього 28 інтерфейсів і відповідних подій, які було згрупувано в 10 категорій по функціональності.

Детально описано основні інтерфейси:

- Головні об'єкти визначення аудіо
 - AudioContext

Представляє собою якийсь аудіо обробний об'єкт, споруджений з аудіо модулів зв'язаних разом, де кожен є екземпляром класу AudioNode. AudioContext управляє створенням вузлів, що містяться в ньому і виконує обробку або декодування аудіо даних.

- AudioNode

Представляє собою обробляючий модуль такий, як джерело аудіо (тобто HTML <audio> або <video> елемент), кінцевий аудіо об'єкт, модуль безпосередньої обробки.

- AudioParam

Представляє аудіо-параметри пов'язані з `AudioNode`. Він може містити як точне значення параметра, так і параметри, які змінюються в часі.

- `ended (en-US) (event)`

Подія `ended` генерується тоді, коли відтворення зупинено через закінчення носія.

- Джерела звуку
- `OscillatorNode(en-US)`

Представляє собою джерело синусоїдальної хвилі. Це модуль `AudioNode` який приймає на вхід частоту і генерує синусоїдальну хвилю з цією частотою.

- `AudioBuffer`

Інтерфейс `AudioBuffer` є коротким аудіо семплом, знаходиться в пам'яті, створений з аудіо файлу з використанням методу `AudioContext.decodeAudioData()`, або створений з необроблених (сирих) даних з використанням методу `AudioContext.createBuffer()` (`en-US`). Після декодування аудіо дані можуть бути поміщені в `AudioBufferSourceNode (en-US)`.

- `MediaElementAudioSourceNode`

Представляє джерело звуку, що міститься в елементі HTML5 `<audio>` або `<video>`. Це `AudioNode`, який діє в якості джерела звуку.

- `MediaStreamAudioSourceNode(en-US)`

Представляє джерело звуку, що міститься в потоці WebRTC `MediaStream` (наприклад веб-камери або мікрофона). Це `AudioNode`, який діє в якості джерела звуку.

- Аудіо фільтри
- `BiquadFilterNode(en-US)`

Представляє простий фільтр низького порядку. Це `AudioNode` представляє різні види фільтрів такі як еквайзер або управління тональністю. `BiquadFilterNode` має рівно один вхід і один вихід.

- `ConvolverNode(en-US)`

Це `AudioNode` виконує лінійну згортку `AudioBuffer`, часто застосовується для досягнення ефекту реверберації по заданій імпульсній характеристиці.

- DelayNode(en-US)

Представляє собою лінію затримки (delay-line).

- DynamicsCompressorNode(en-US)

Інтерфейс забезпечує ефект стиснення, який зменшує гучність найсильніших частин сигналу, щоб запобігти спотворенням, які можуть виникнути при відтворенні та мультиплексуванні кількох звуків одночасно.

- GainNode(en-US)

Інтерфейс представляє зміну гучності. Це модуль обробки звуку AudioNode, який змушує застосовувати даний коефіцієнт посилення до вхідних даних перед його розповсюдженням на вихід.

- StereoPannerNode(en-US)

Представляє собою простий стереовузол панорами, який можна використовувати для панорамування аудіопотоку вліво або вправо.

- WaveShaperNode(en-US)

Представляє собою нелінійний спотворювач.

- PeriodicWave(en-US)

Використовується для визначення періодичної форми сигналу, яка може бути використана для формування вихідних даних OscillatorNode (en-US).

- Призначення аудіо

Після завершення обробки звуку ці інтерфейси визначають, куди його виводити.

- AudioDestinationNode(en-US)

Інтерфейс AudioDestinationNode являє собою кінцевий пункт призначення джерела звуку в певному контексті — зазвичай це динаміки пристрою.

Інші існуючі інтерфейси:

- Аналіз та візуалізація даних
- Розбиття та об'єднання аудіоканалів
- Простір аудіо
- Обробка звуку за допомогою JavaScript

- Офлайн / фонові обробка звуку
- AudioWorkers

3.3. Створення веб-сайту.

3.3.1. Вибір середовища для програмування.

Для створення проекту використано програму Visual Studio Code.

Згідно з джерелом [14], Visual Studio Code — засіб для створення, редагування сучасних веб-застосунків і програм для хмарних систем. Редактор містить вбудовані інструменти для роботи з Git і засоби рефакторингу, навігації по коду, автодоповнення типових конструкцій і контекстної підказки.

Створення проекту. При першому відкритті програми відображається початкове вікно, де можна натиснути «Створити файл» (або комбінація клавіш Ctrl + N).

3.3.2. Додавання React.

Так як було зазначено раніше, писати код ми будемо за допомогою бібліотеки React, скористаємось готовим шаблоном стандартного React-додатку. Для цього в консолі потрібно прописати наступну команду:

```
npx create-react-app «назва проекту»
```

Після установки, структура нашого проекту буде виглядати як показано на Рис. 3.2

Опис структури проекту:

- public — статичні файли, які не компілюються React, у файлі index.html точка входу додатку. Тут зазвичай зберігаються всі статичні файли: картинки, шрифти, іконки та інше;
- src — Вся логіка додатку на React;
- package.json — це файл, який зберігає інформація про створений додаток: назва, версія, npm-пакети (розширення) і скрипти(для запуску програми, тестування та інше);
- src / index.js — головний файл react-додатку, вся логіка починається з нього;

- App.js — наш перший компонент, до нього будуть приєднуватись всі інші компоненти.

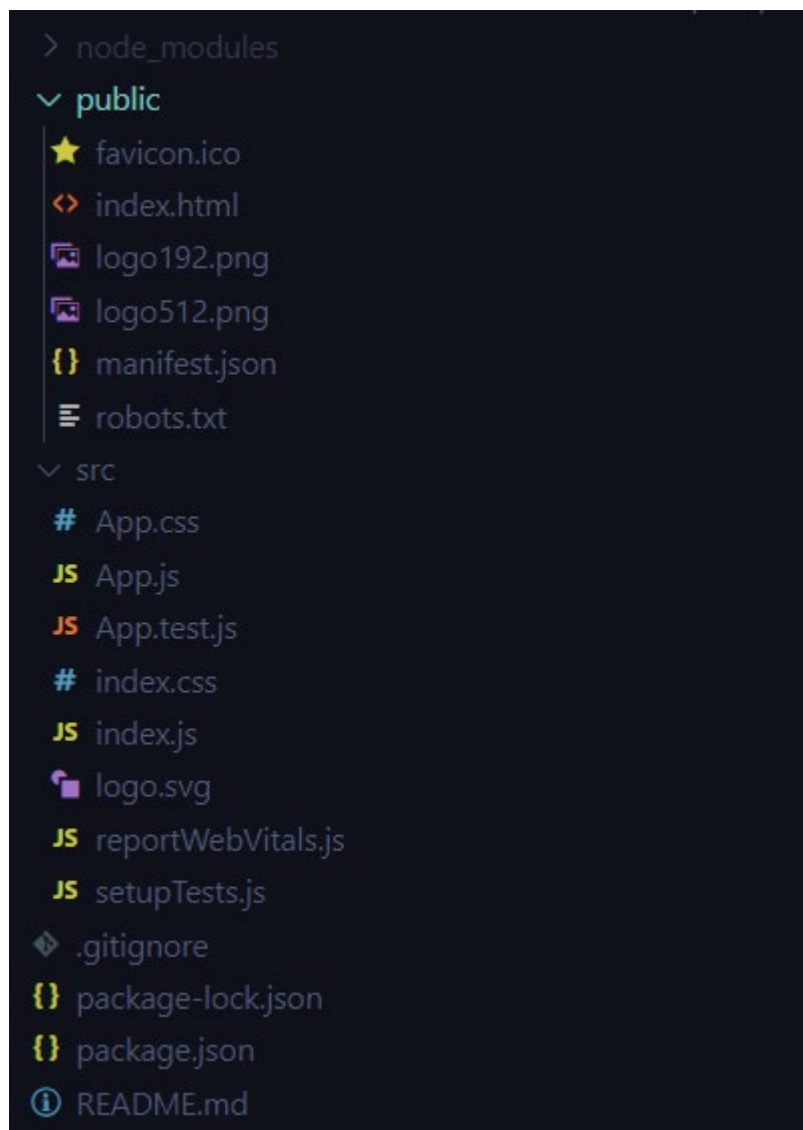


Рис. 3.2 Структура створеного проекту

Короткий опис як працює стандартний React-додаток:

Для початку, подивимося першу точку входу. Це файл `public / index.html`, він являв собою простий html-документ, але є невелика особливість, в ньому є один цікавий рядок:

```
<div id="root"></div>
```

За допомогою цього рядка(тега) буде працювати весь додаток.

Далі, у директорії `src/index.js` — наш головний файл React-додатку, тут підключається сам `react`, `react-dom` (для роботи з web-додатками), підключаємо

стартовий компонент App та вказуємо в якому тезі з файлу public/index.html буде працювати наш додаток.

```
ReactDOM.render(, document.getElementById('root'));
```

Тут ми бачимо, що компонент App, який вже підключений буде оброблятися в елементі з id = "root" з файлу public/index.html.

А в файлі App.js — наш перший компонент, який повертає просту html-розмітку.

3.3.3. Створення компонентів.

Компоненти — це незалежні функціональні елементи, які можна багаторазово використовувати в додатках. Всі додатки React будуються на основі компонентів. Кнопки, меню і будь-які інші матеріали на головній сторінці, які призначені для користувача інтерфейсу можна створювати як компонент. [15]

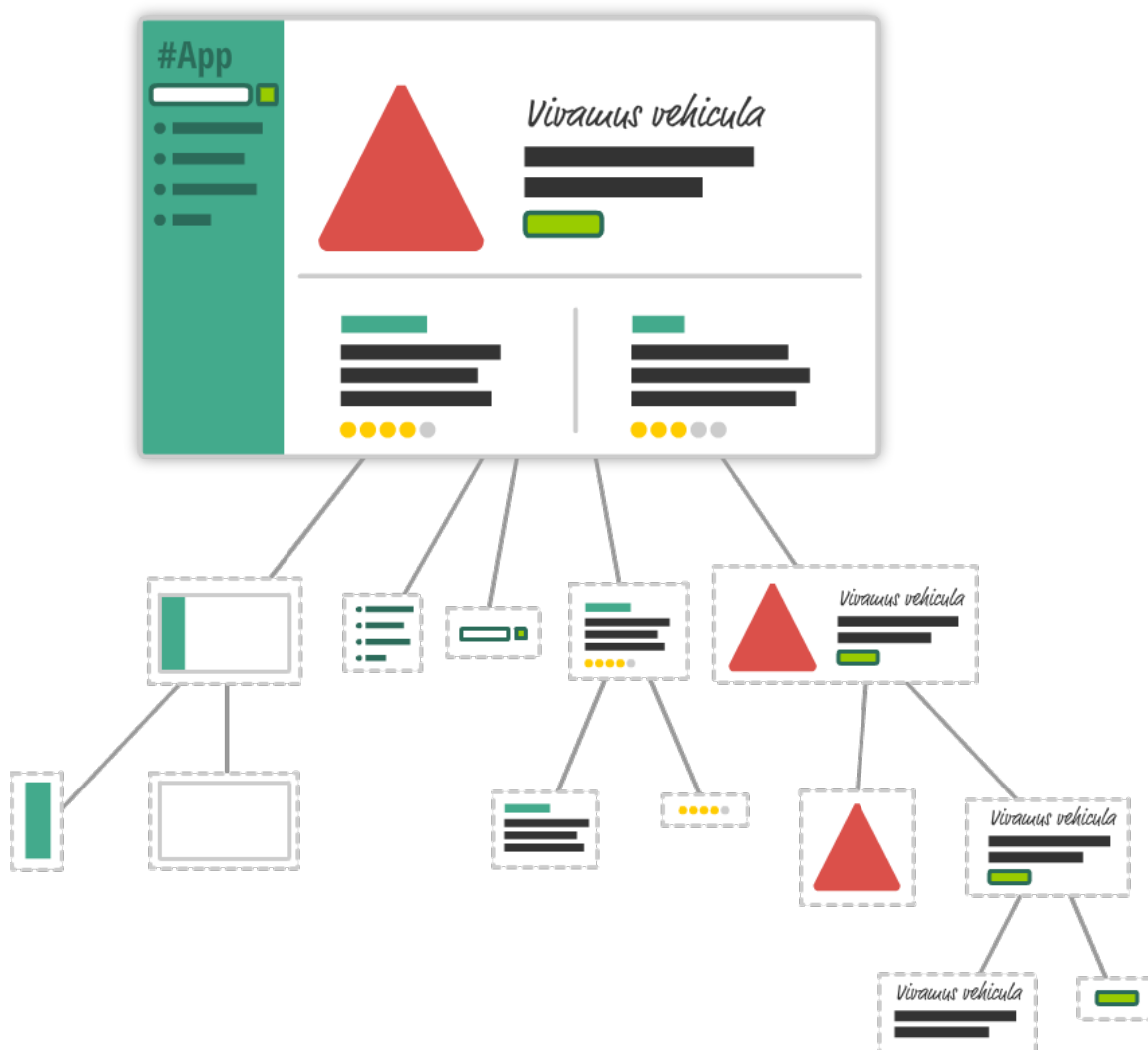


Рис. 3.3 Приклад веб-застосунку, який поділений на компоненти [16]

Всі компоненти зберігаються у директорії src/components.

Список компонентів, які використанні при створенні сайту:

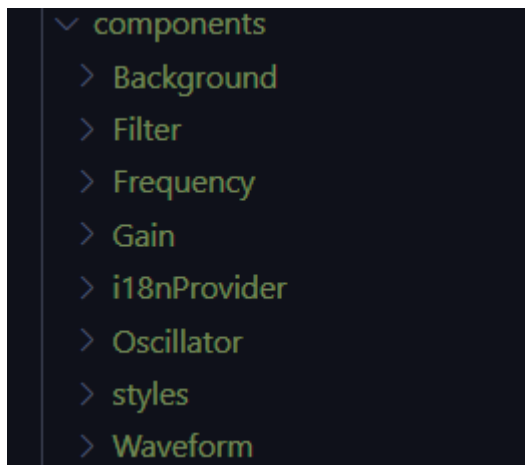


Рис. 3.4 Скріншот папок створених компонентів

Опис компонентів:

1. Background – компоненти відповідає за задній фон на веб-сайті.
2. Filter – компонент відповідає за вибір фільтру. На сайті є можливість вибрати один з фільтрів, як показано на Рис. 3.5

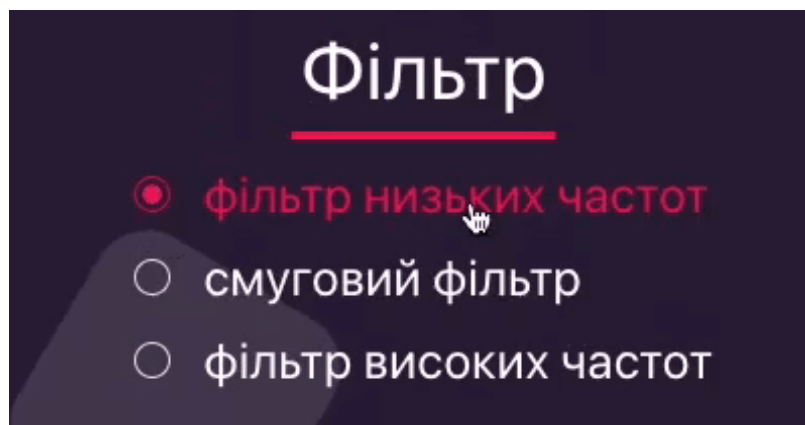


Рис. 3.5 Вигляд вибору фільтра на сайті

3. Frequency – компонент відповідає за регулювання частоти. Ось такий вигляд на сайті, як показано на Рис. 3.6

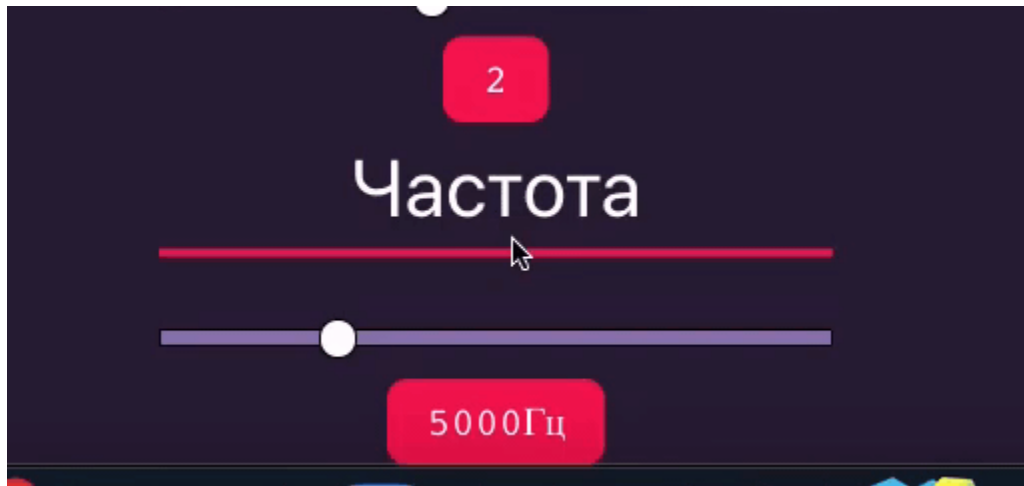


Рис. 3.6 Вигляд налаштування частоти на сайті

4. Gain – компонент відповідає за регулювання звуку, як показано на Рис. 3.7.

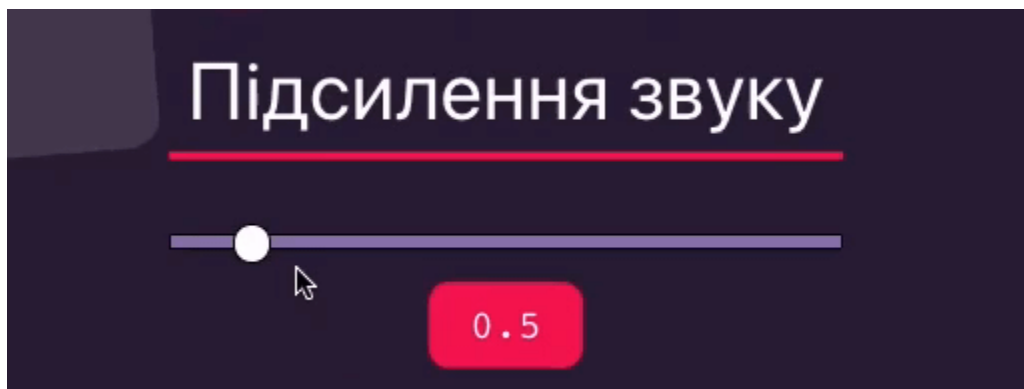


Рис. 3.7 Вигляд налаштування звуку на сайті

5. i18nProvider – компонент відповідає за інтернаціоналізацію сайту.

За допомогою бібліотеки react-intl [17] зроблено інтернаціоналізацію сайту. Ця бібліотека надає можливість створення «словника». У цій директорії є папка з файлами-словниками: en-US.js, ru.js, uk-UA.js

В них написані слова, які потрібно підставити на різні місця на сайті.

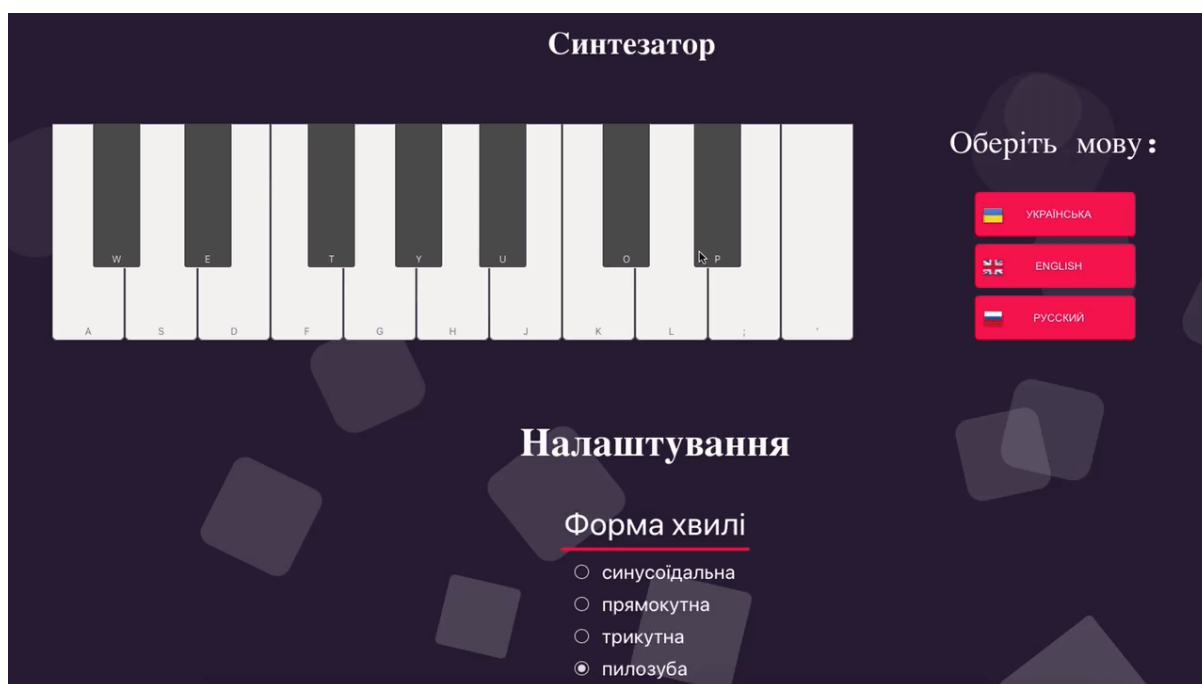


Рис. 3.8 Вигляд вибору мови на сайті

6. Oscillator – компонент відповідає за клавіатуру та інтерфейс OscillatorNode.
7. Waveform – компонент відповідає за вибір форми хвилі.

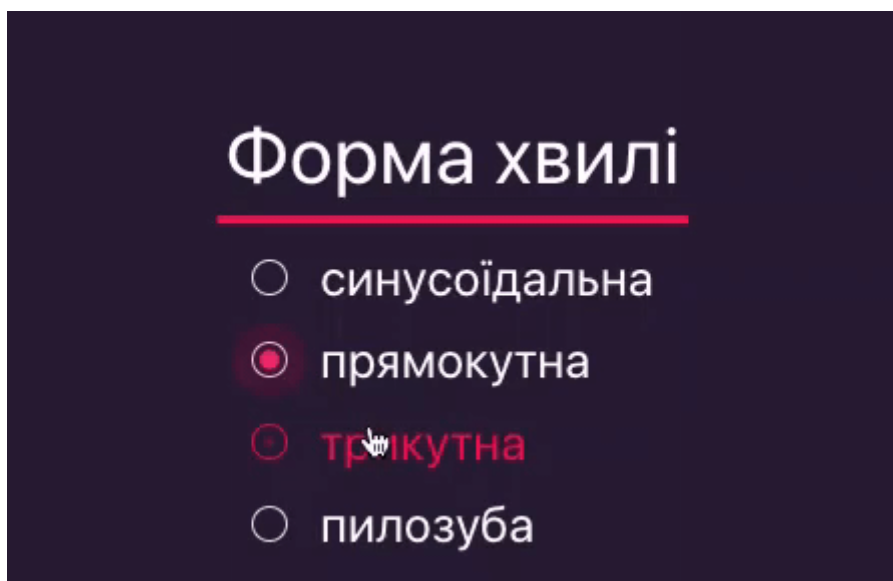


Рис. 3.9 Вигляд вибору форми хвилі на сайті

8. styles – це проста директорія зі спільними стилями для компонентів.

3.3.4. Створення синтезатора

- Клавіатура

Найголовніше, що потрібно синтезатору, це клавіатура.

Її було створено за допомогою бібліотеки `react-piano` [18]. Це інтерактивна клавіатура для React. Підтримує власні звуки, дотики / клацання / події клавіатури та налаштований стиль.

- Осцилятор

Для початку потрібно створити аудіо контекст:

```
let AudioContext = new (window.AudioContext || window.webkitAudioContext);
```

Цей варіант створення контекста є кросбраузерним.

Як вказано в джерелі [19], кросбраузерність – це властивість сайту однаково відображатися та функціонувати у відповідності до поставленого завдання в усіх браузерах.

Всі терміни використані відповідно до [1].

Короткий опис методів інтерфейсу `AudioContext`, які були використанні при написанні програми:

- `AudioContext.createBiquadFilter()`

Створює об'єкт `BiquadFilterNode` (en-US), який представляє собою фільтр високих/низьких частот і смуговий фільтр.

- `AudioContext.createGain()`

Створює об'єкт `GainNode` (en-US), який може бути використаний для контролю загальної гучності створеного пристрою.

- `AudioContext.createOscillator()`

Створює об'єкт `OscillatorNode` (en-US), джерело, що представляє собою осцилятор.

- `AudioContext.destination`

Містить посилання на `AudioDestinationNode` (en-US), що представляє собою точку призначення для всього аудіо в цьому контексті. Може розглядатися як, наприклад, динамік комп'ютера.

- `AudioContext.currentTime`

Містить значення часу в секундах, отримане від годинника комп'ютера користувача. Це значення не може бути змінено ніяким чином, воно тільки для читання.

У компоненті `Oscillator` є дві основні функції `playSound` та `stopSound`.

Функція `playSound` відповідає за відтворення звуку. Саме у ній створено осцилятор.

В рамках готового аудіо контексту було створено осцилятор.

```
let oscillator = audioContext.createOscillator();
```

Електронний осцилятор — електричний генератор, який виробляє повторювальний електронний сигнал.

За допомогою `createOscillator` створено інтерфейс `OscillatorNode`.

Властивості інтерфейсу:

- `OscillatorNode.frequency`

Представляє собою об'єкт `AudioParam`, який вказує параметр частоти коливань звукової хвилі в герцах.

- `OscillatorNode.detune`

Представляє собою об'єкт `AudioParam`, який вказує параметр `detune` в центах.

- `OscillatorNode.type`

Вказує параметр типу форми звукової хвилі на виході.

Типи можливих форм хвилі:

`sine` — синусоїда. Це значення за замовчуванням.

`square` — квадратна хвиля з робочим циклом 0,5.

`sawtooth` — пилоподібна хвиля.

`triangle` — трикутна хвиля.

Методи:

- `OscillatorNode.start()`

Вказує точний час початку відтворення звуку.

- `OscillatorNode.stop()`

Вказує час зупинки відтворення звуку.

Задамо осцилятору частоту, підключимо до аудіовиходу і потім змусимо відтворювати звук.

Для створення регулятора гучності та вибору фільтру, напишемо наступне:

```
let gainNode = audioContext.createGain();  
let biquadFilter = audioContext.createBiquadFilter();
```

для підключення осцилятора до аудіо входу використовується:

```
audioContext.destination();
```

Функція `stopSound` відповідає за зупинку відтворення звуку. Це робиться за допомогою функції `oscillator.stop()`

3.4. *Запуск програми*

Щоб запустити програми потрібно зробити наступні кроки:

1. Переходимо у термінал
2. Вводимо команду `npm start`

Також сайт можна викласти на платний або безкоштовний хостинг.

Висновки

У роботі вирішено актуальне завдання — розробка віртуального синтезатора за допомогою мови програмування JavaScript та програмного інтерфейсу Web Audio API. В ході виконання даної роботи було проведено дослідження у сфері синтезу звуку, розглянуто різні типи синтезу звуку. Зробивши порівняння реального синтезатора з програмним, можна зробити наступний висновок. Кожен синтезатор має свої плюси і мінуси, віртуальний виграє доступністю, а реальний якістю і кількістю налаштувань.

Ця робота представляється, як рішення для дистанційного навчання. Зроблено веб-сайт віртуального синтезатора з стандартними налаштуваннями. З плюсів даної роботи можна виділити такі функції, як наявність 3 мов(англійська, російська, українська), на відміну від конкурентів це безкоштовний ресурс, можливість удосконалення.

Посилання на сайт: <https://synthesizerdiploma.netlify.app/>

Перелік посилань

- [1] Mozilla and individual contributors, «Web Audio API,» developer.mozilla.org, [В Інтернеті]. Available: https://developer.mozilla.org/ru/docs/Web/API/Web_Audio_API. [Дата звернення: 17 04 2021].
- [2] Wikimedia Foundation, Inc., «Комп'ютерна музика,» [В Інтернеті]. Available: https://ru.wikipedia.org/wiki/Компьютерная_музыка. [Дата звернення: 15 05 2021].
- [3] А. Шугаев, «Развитие компьютерных технологий в музыке XX века,» [В Інтернеті]. Available: <https://samesound.ru/music/5032-razvitie-kompyuternyh-tehnologij-v-muzyke-xx-veka>. [Дата звернення: 15 03 2021].
- [4] StudFile, «Методы синтеза звука,» studfile.net, [В Інтернеті]. Available: <https://studfile.net/preview/6062586/page:4/>. [Дата звернення: 24 03 2021].
- [5] Інтернет-магазин музичних інструментів Музлайн, «Синтезатори для навчання,» muzline.ua, [Онлайнний]. Available: <https://muzline.ua/syntezaory-dlya-obuchenyaya/>. [Дата звернення: 15 05 2021].
- [6] Інтернет-магазин «Розетка™», «Синтезатор Casio SA-47,» <https://rozetka.com.ua/>, [Онлайнний]. Available: https://rozetka.com.ua/casio_sa_47/p305633/characteristics/. [Дата звернення: 15 05 2021].
- [7] B. Sievers, “Synthesis Basic,” [Online]. Available: <http://beausievers.com/synth/synthbasics/>. [Accessed 12 03 2021].
- [8] Wikimedia Foundation, Inc., «Форми хвилі,» [Онлайнний]. Available: https://uk.wikipedia.org/wiki/Хвиля#Форми_хвилі.. [Дата звернення: 10 03 2021].
- [9] Jetronic Music, «Основы сведения и мастеринга,» [В Інтернеті]. Available: <https://stereo-max.blogspot.com/2019/07/adsr.html>. [Дата звернення: 12 03 2021].
- [10] Wikimedia Foundation, Inc., «Синтезатор,» [Онлайнний]. Available: <https://uk.wikipedia.org/wiki/Синтезатор>. [Дата звернення: 10 03 2021].

- [11] Encyclopædia Britannica, Inc., "Digital sound recording," britannica.com, [Online]. Available: <https://www.britannica.com/technology/digital-sound-recording>. [Accessed 12 03 2021].
- [12] Center Point Audio, "Understanding the difference between Analog and Digital Audio," [Online]. Available: <http://www.centerpointaudio.com/Analog-VS-Digital.aspx>. [Accessed 12 03 2021].
- [13] Wikimedia Foundation, Inc., «Прикладний програмний інтерфейс,» [Онлайновий]. Available: https://uk.wikipedia.org/wiki/Прикладний_програмний_інтерфейс. [Дата звернення: 27 04 2021].
- [14] Wikimedia Foundation, Inc., «Visual Studio Code,» [Онлайновий]. Available: https://uk.wikipedia.org/wiki/Visual_Studio_Code. [Дата звернення: 29 04 2021].
- [15] J. Morgan, «Создание пользовательских компонентов в React,» digitalocean.com/, [В Интернете]. Available: <https://www.digitalocean.com/community/tutorials/how-to-create-custom-components-in-react-ru>. [Дата обращения: 02 05 2021].
- [16] "Introducing React," [Online]. Available: https://www.kirupa.com/react/introducing_react.htm. [Accessed 02 05 2021].
- [17] npm, Inc, "React Intl," npmjs.com, [Online]. Available: <https://www.npmjs.com/package/react-intl>. [Accessed 25 04 2021].
- [18] npm, Inc, "react-piano," npmjs.com, [Online]. Available: <https://www.npmjs.com/package/react-piano>. [Accessed 23 04 2021].
- [19] Wikimedia Foundation, Inc., «Кросбраузерність,» [Онлайновий]. Available: <https://uk.wikipedia.org/wiki/Кросбраузерність>. [Дата звернення: 29 04 2021].

ДОДАТКИ

Додаток А. Лістинг програми

Весь лістинг програми знаходиться на електронному ресурсі:
<https://github.com/AlinaTaran/diploma>