

РАДІОТЕХНІКА ТА ТЕЛЕКОМУНІКАЦІЇ

УДК 681.3.07

DOI <https://doi.org/10.32838/2663-5941/2019.6-1/04>

Борисов Г.О.

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

Трапезон О.Г.

Інститут проблем міцності імені Г.С. Писаренка Національної академії наук України

Трапезон К.О.

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

ОСОБЛИВОСТІ ОБРОБЛЕННЯ АКУСТИЧНИХ СИГНАЛІВ ІЗ ЗАСТОСУВАННЯМ ПРОГРАМНИХ ЗАСОБІВ IOS

Наведено опис головних можливостей операційної системи iOS та проведено аналіз особливостей функціонування даної інформаційної платформи. Розглянуто структуру базової модульної архітектури мобільного додатку, який може бути побудований на основі правил і процедур зазначеної операційної системи. Okремо визначено, що шаблон для створення додатку визначається особливостями об'єктно-орієнтованого програмування і складається щонайменше з трьох основних компонентів, де самі дані, які завантажуються для аналізу чи обробки розташовані, наприклад, в модулі «модель» архітектури мобільного додатку. Додатково зазначено, що цей складник архітектури мобільного додатку відокремлений від даних, пов'язаних із візуальним оформленням додатку, а отже будь-які зміни в значеннях змінних чи посиланнях не будуть безпосередньо впливати на інтерфейс створеного додатку.

Визначено основні принципи зі створення мобільного додатку та розроблено програмний код кадру, класу, методу через реалізацію процедури розкадрування додатку. Знайдено, що така процедура визначається на основі навігаційного шаблону, який через програмні засоби можна описати у формі стандартного навігаційного контролеру.

Побудовано алгоритм дій зі створення простого вузлового програвача акустичних сигналів. Проведено тестування можливостей створеного додатку на основі доданих різних методів цифрової обробки акустичних сигналів. Так, в межах створеного додатку проведено тестування двох програмних контролерів, які дозволяють не лише відтворити акустичний сигнал з певною тривалістю та акустичними характеристиками, але й у користувача при цьому з'являється можливість при такій структурі провести цей сигнал через доступний набір ефектів та фільтрів.

Визначено, що структура взаємодії програмних контролерів визначається через алгоритм з послідовними зв'язками на основі з'єднаних між собою вузлів (нод). При цьому, доступ до окремого акустичного ефекту в такій структурі реалізовано на основі сигналів коду від навігаційного контролеру мобільного додатку.

Ключові слова: додаток, платформа, фреймворк, зображення, акустичний сигнал, алгоритм, обробка.

Постановка проблеми. Розроблення мобільних додатків у розрізі стрімкого розвитку інформаційних технологій все частіше потребує від проектувальників застосування нових підходів, які нерозривно пов'язані сьогодні з програмуванням та тестуванням створених додатків. Одним із проявів таких підходів можна вважати операційну систему iOS та застосування, які працюють на її

основі. Разом з тим, способи та алгоритми, які лягли в основу функціонування сучасних мобільних гаджетів та пристроїв американської фірми Apple завдяки відкритості цих алгоритмів, можна просто дослідити та модернізувати на основі нових розробок в сфері телекомунікаційних технологій. Через це розробки можна не лише прописати у кодї в певному програмному середовищі,

але й оформити у вигляді додатку, який дозволить, наприклад, провести цифрову обробку різних інформаційних сигналів. До числа цих сигналів можна віднести звукові сигнали, а в якості відображення інформації можна обрати зображення певного формату.

Постановка завдання. Система iOS – обчислювальна платформа і разом перспективне сучасне середовище для розробки різних мультимедійних додатків. Використовуючи ресурси цієї платформи у користувача мобільного гаджету з'являються по суті необмежені можливості зі створення різних звукових сигналів, які, наприклад, можуть повністю замінити звуки природи, звуки тварин а їх обробка може стати основою до створення різних спеціальних звукових ефектів, звукових фільтрів та відповідних інструментів. Перевагою такого підходу є прозорість усіх дій з звуковою обробкою, адже код, спеціальна мова програмування, спеціальні аудіо модулі є відкритими в середовищі мережі Інтернет, і розробники власноруч на основі засобів тестування можуть провести цілий ряд експериментів з обраним звуковим семплом. Фактично, на основі інформаційної платформи iOS та засобів запису, прослуховування та відповідних бібліотек можливо забезпечити корегування параметрів наявних звукових сигналів, не лише через стандартні звукові ефекти, але й шляхом внесення змін у записані сигналограми.

Метою статті є дослідження принципів створення мобільного додатку на основі інтегрованого середовища xCode в рамках операційної системи iOS задля проведення практичного експерименту з обробки обраних звукових сигналів та певних цифрових зображень з стандартною чіткістю як основних елементів мультимедійного контенту.

Виклад основного матеріалу дослідження. По суті, всі додатки чинної версії операційної системи iOS використовують архітектуру «Модель – вид – контролер» (Model-View-Controller). Модель, вид і контролер – це три основних компоненти додатку (з точки зору його архітектури) в операційній системі iOS [1–2]. Model-View-Controller (MVC) – це надзвичайно потужний програмний архітектурний шаблон для створення додатків. MVC, скорочення від Model-View-Controller, є програмним архітектурним патерном. Model-View-Controller заснований на об'єктно-орієнтованому програмуванні. Він структурує потік даних і взаємодію в додатку. На рисунку 1 наведена структура MVC. Модель (Model) містить основні дані, наприклад змінні, підключення

до зовнішніх RSS-каналів або зображень, функції і числову інформацію. Цей шар повністю відділяється від візуального оформлення, і через це можна легко змінити вигляд дисплея, і при цьому це не вплине на самі дані.

Вид (View) відповідає за стиль відображення інформації на дисплеї. В якості уявлення виступає екран з графічними елементами, на якому, наприклад, можна змінити стиль або видаляти різні елементи. Контролер (Controller) управляє запитами користувача і використовує модель для реалізації необхідної реакції. Модель і вид ніколи не контактують один з одним. Вони не повинні навіть знати один про одного в ідеальному випадку [3–4].

Вид несе відповідальність за відображення свого користувацького інтерфейсу для користувача. Зрештою, уявлення передає дії, створені з користувацького, в контролер. Дії або ініціюються користувачем, наприклад, користувач може натиснути на кнопку, присутню на екрані, щоб ініціювати дію, або іноді автоматично генеруються, наприклад, екран може потребувати своєчасному оновлення свого вмісту. Відповідальність контролера полягає в тому, щоб отримувати ці дії від виду і відповідати на них. Контролер під час обробки може оновити модель. Іноді цей процес оновлення займає деякий час. Модель повідомляє контролер про завершення процесу оновлення, після чого контролер отримує повідомлення про оновлення свого користувацького інтерфейсу оновленими даними. Вид і модель ніколи не спілкуються один з одним безпосередньо. Вони спілкуються через контролер. Компанія Apple відхилилась від стандартного шаблону MVC. Вони прийняли рішення про те, що не слід тісно пов'язувати уявлення з моделлю тільки через те, що подання до мобільних додатках можна багаторазово використовувати. Якби зберігався тісний зв'язок, довелося б кожен раз заново реалізовувати уявлення для різних моделей.

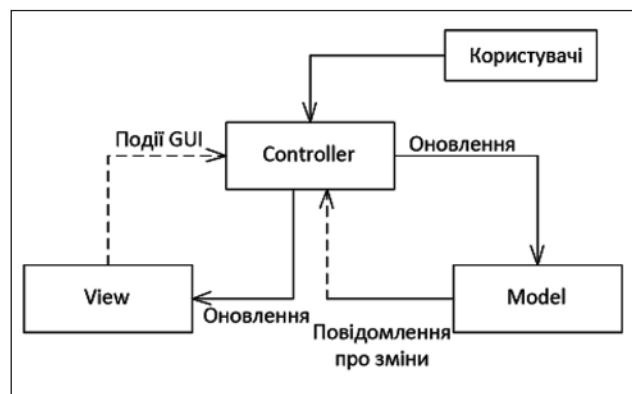


Рис. 1. Архітектура Model-View-Controller

Разом з тим, модель незалежного уявлення – актуальна ідея MVC Apple. Перевага цієї модульної архітектури в тому, що поділ ролей дозволяє нам легко вносити зміни з меншою кількістю помилок. Якщо в майбутньому вам потрібно буде змінити спосіб вибірки або організації даних, все, що вам потрібно зробити, це перемкнути модель.

Алгоритм побудови мобільного додатку

Побудова будь-якого складного додатку починається з розкадрування. Розкадрування – це кількість окремих самостійних сцен (кадрів) додатку, які пов’язані між собою. Розкадрування складається з головного виду (Main View Controller), який завантажується при запуску додатку, та дочірніх видів. Головний вид має посилання на дочірні види. Зворотні посилання можна створити самостійно, використавши один із навігаційних шаблонів або вказавши, що вид являється контролером навігації. Для створення навігаційного контролера треба виділити стандартний контролер, навігаційний контролер (NavigationController) та зв’язаний із ним кадр (рис. 2), на якому розміщується мультимедійний контент.

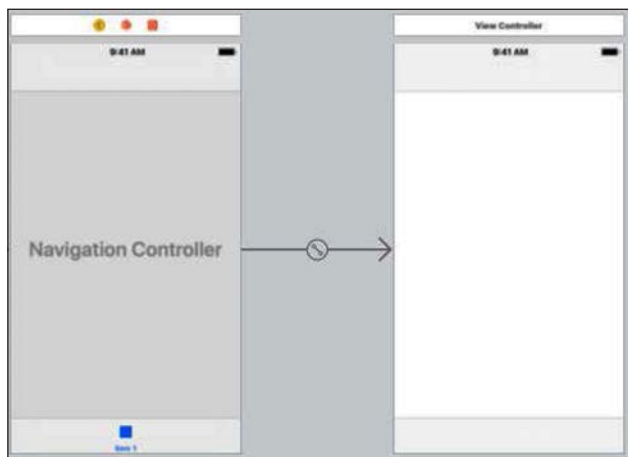


Рис. 2. Навігаційний контролер та контролер виду

З головного виду знадобилось 2 послання на кадр, який відповідає за взаємодію за аудіо файлами, та на кадр, який відповідає за взаємодію із зображеннями. В якості головного виду був використаний об’єкт Tab Bar Controller, який автоматично генерує 2 дочірніх кадра та посилання на них в нижній частині. Tab Bar Controller працює не як стандартний навігаційний контролер, він змінює своє наповнення залежно від обраного посилання, а не переходить на них, за замовчування завантажується, та яка належить першому посиланню з права. Перше посилання Audio направляє до навігаційного контролера та зв’язаного з ним виду Audio які відносяться до гілки яка відповідає за аудіо контент. Друге посилання Img направляє до виду Img, який належить до гілки розкадрування

яка відповідає за обробку зображень. Вид Img є кінцевий та наповнений зображенням, на яке накладаються ефекти та 6 кнопок, що відповідають за обрані ефекти. Вид Audio має два посилання Player та Autotune: перше веде до кінцевого кадру Player, який відповідає за програвання аудіо-файлу та містить зображення та три кнопки: почати, зупинити та повторити з початку. Друге посилання веде до кінцевого кадру ViweWithTableList, який відповідає за обробку аудіо. Він містить контейнер для комірок (Table View), у якому розміщена одна комірка (Table View Cell) з назвою Prototype Cell та дві кнопки: почати та зупинити. При завантаженні додатку Table View повністю заповнюється комірками, кожна з яких відповідає за застосування фільтру або аудіо ефекту до звукової доріжки. Комірка має два поля: одне поле для назви ефекту, друге – для відображення в цифровому вигляді частоту фільтру, процент накладання певного ефекту тощо. Також комірка має повзунок, за допомогою якого регулюються параметри ефектів для накладання. Інформацію якою наповнюється комірка описує клас Effect. Клас Effect несе в собі наступну інформацію: назва ефекту filter, мінімальні та максимальні значення повзунка min та max, початкове положення value та колір поля Name. За наповнення комірки інформацією відповідає клас AvtotyunCell. В рамках побудови мобільного додатку для програвання звукових файлів проведено створення 10 екземплярів класу Effect, кожен з яких відповідає за певний ефект, але всі вони поміщаються в масив tempEffects. При виклику функції, вона повертає масив tempEffects, інформація якого присвоюється масивом effects, для подальшого створення і наповнення комірок. Повний вигляд розкадрування зображений на рисунку 3.

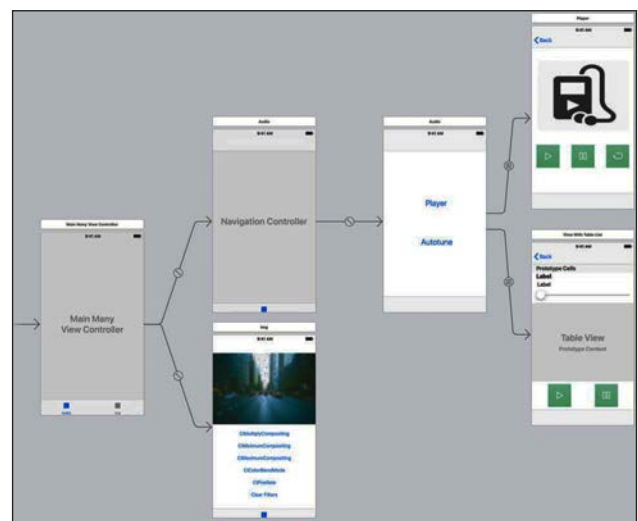


Рис. 3. Розкадрування додатку

Схема тестування

Обробка аудіофайлів здійснюється за допомогою двох програмних контролерів: Player та ViewWithTableList. Контролер Player виконує функції плеєра. Він надає можливості відтворювати, зупиняти та програвати заново аудіо. Контролеру Player належить кадр із такою ж назвою Player. Взаємодія з аудіо-доріжкою здійснюється за допомогою фреймворку AVFoundation. Для реалізації функціоналу плеєра було використано клас AVAudioPlayer, який містить в собі всі необхідні інструменти для цієї задачі. Спочатку створюється об'єкт player класу AVAudioPlayer, та при ініціалізації слід вказати йому шлях до аудіо файлу, який буде використовуватися.

Накладання різних ефектів та фільтрів реалізується дещо складнішим способом, ніж просто відтворення та зупинка аудіо. В додатку реалізовано 5 звукових ефектів та 5 фільтрів, кожному з яких відповідає своя комірка, в якій налаштовуються параметри звукового ефекту чи фільтру:

- Pitch – накладає ефект підвищеного чи пониженого тону;
- Reverb – накладає ефект відлуння;
- SpeechRadioTower – накладає ефект “SpeechRadioTower”;
- MultiDecimated1 – накладає ефект “MultiDecimated1”;
- MultiDecimated2 – накладає ефект “MultiDecimated2”;
- BandPass – смуговий фільтр;
- LowPass – фільтр нижніх частот;
- HighPass – фільтр високих частот;
- BandStop – режекторний фільтр;
- LowShelf – низькочастотний шельфовий фільтр.

Кожен ефект та фільтр скріплюються між собою вузлом для програвання аудіо, мікшером та вузлом, який відповідає за вивід звуку. Всі вони є аудіо вузлами. Блок-схема зв'язку аудіо вузлів наведена на рисунку 4.

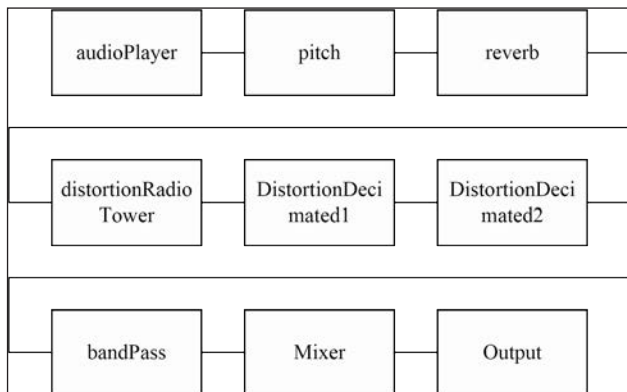


Рис. 4. Схема зв'язку аудіо вузлів

При тестуванні додатку було використано акустичний сигнал тривалістю 1 хвилина, сигналограма якого у формі послідовних відліків наведена на рисунку 5. Саме на неї накладаються вибрані ефекти та фільтри.

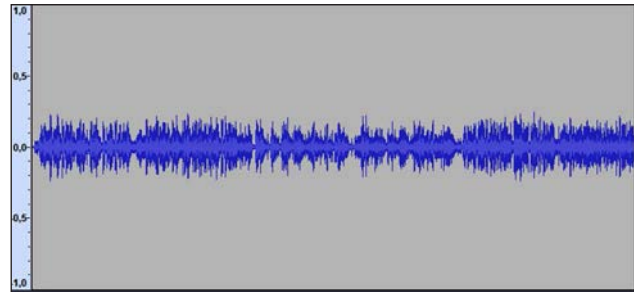


Рис. 5. Тестова сигналограма сигналу

На тестову аудіо доріжку для накладання було обрано ефект Reverb (ефект реверберації), який емітує відлуння всередині середовища собору, як приміщення з певними акустичними характеристиками. Також поверх ефекту відлуння було накладено фільтр LowPass (фільтр низьких частот) з частотою зрізу 1446 Гц (рисунк 6).

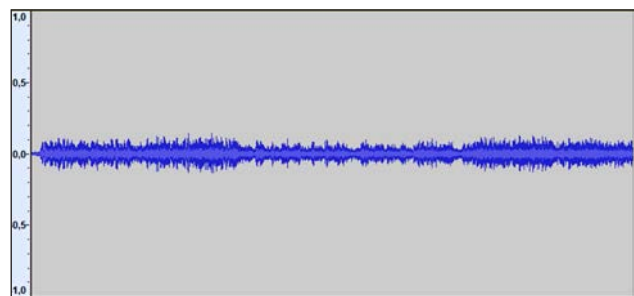


Рис. 6. Тестова сигналограма з накладеним ефектом відлуння та фільтром LowPass

На основі проведеного дослідження з перевірки можливостей створеного мобільного додатку, можна підкреслити, що хвильова форма, до якої було застосовано послідовно два акустичних ефекти, визначається тим, що енергетичний складник сигналу внаслідок акустичної обробки значно зменшився, що обумовлено акустичними особливостями віртуального приміщення-собору. При цьому, тональні складові сигналу для обраного семплу відповідають природі у випадку обробки сигналів, яку можна провести в спеціалізованих програмних комплексах.

Висновки. Побудовано алгоритм дослідження найбільш розповсюджених звукових ефектів на основі використання можливостей інформаційної платформи iOS. Описано архітектуру «Модель-вид-контролер» у формі компонентів додатку, що є базовою основою до проведення експерименту

з аналізу особливостей мультимедійного контенту на основі використання мобільних гаджетів та спеціальної мови програмування.

Побудовано простий алгоритм використання спеціальних фреймворків в середовищі XCode, який дозволив провести тестування на обраному семплі ряд звукових ефектів. Знайдено, що побудований алгоритм аналізу семплів на основі ресурсів iOS дозволяє значно спростити сам процес дослідження звукових сигналів різ-

ної природи, через те, що інструменти симулятора описуються через код мови програмування, і отже їх налаштування є зрозумілим підходом до проведеного аналізу. Крім цього, наведена схема підключення вузлів в середовищі Xcode обчислювальної платформи iOS може бути використана для додавання власних звукових ефектів між елементами програвача та виходу, та на різних етапах конструювання віртуальних генераторів звуку.

Список літератури:

1. Нахавандипур В. iOS. Приемы программирования. Санкт-Петербург : «Питер», 2014. 815 с.
2. Усов В. Swift. Основы разработки приложений под iOS и macOS, 3-е изд., доп. и перераб. СПб : Питер, 2017. 368 с.
3. Маскри М., Топли К., Марк Д. Swift 3: разработка приложений в среде Xcode для iPhone и iPad с использованием iOS SDK, Санкт-Петербург : «Питер», 2017. 896 с.
4. В. Нахавандипур iOS. разработка приложений для iPhone, iPad и iPod. Санкт-Петербург : «Питер», 2013. 864 с.

Borysov G.O., Trapezon O.G., Trapezon K.O. ACOUSTIC SIGNAL PROCESSING FEATURES BASED ON IOS SOFTWARE

The main features of the iOS operating system are described and the peculiarities of the functioning of this information platform are analyzed. The structure of the basic modular architecture of the mobile application, which can be built on the rules and procedures of the specified operating system, is considered. Specifically, an application creation template is determined by the features of object-oriented programming and consists of at least three major components, where the data that is downloaded for analysis or processing is located, for example, in the "model" module of the mobile application architecture. In addition, it is noted that this component of the mobile application architecture is separate from the data related to the visual design of the application, and therefore any changes in the values of variables or links will not directly affect the interface of the created application.

The basic principles for the creation of a mobile application are defined and the program code of the frame, class, method is developed through the implementation of the application decoding procedure. It is found that such a procedure is based on a navigation template, which through software can be described in the form of a standard navigation controller. An algorithm for creating a simple nodal acoustic signal player has been built. Tested the capabilities of the created application based on the added different methods of digital processing of acoustic signals.

Thus, within the created application, two software controllers were tested, which allow not only to reproduce an acoustic signal with a certain duration and acoustic characteristics, but also the user has the opportunity to structure this signal through an available set of effects and filters. It is determined that the structure of the interaction of software controllers is determined through an algorithm with serial connections based on interconnected nodes (nodes). In this case, access to a separate acoustic effect in such a structure is realized based on the code signals from the navigation controller of the mobile application.

Key words: *application, platform, framework, image, acoustic signal, algorithm, processing.*