

Результати, отримані в цій роботі, будуть корисні у науково-технічній сфері та сфері захисту даних саме тому, що вони дадуть змогу як початківцям, так і професіоналам приховувати будь-які повідомлення без використання інших програмних засобів.

ЛІТЕРАТУРА

1. Конахович Г.Ф. Компьютерная стеганография. Теория и практика / Г.Ф.Конахович, А.Ю.Пузыренко. – К.: МК-Пресс, 2006. – 288с.
2. Грибунин В.Г. Цифровая стеганография / В.Г.Грибунин, И.Н.Оков, И.В.Турицев. – М.: «Солон-Пресс», 2002. – 272с.
3. Домарев В.В. Безопасность информационных технологий. Системный подход / В.В.Домарев. – К.: ООО "ТИД "ДС", 2004. – 992с.

Надійшла до редколегії 29.01.2018.

УДК 004.52

DOI 10.31319/2519-2884.32.2018.176

МІНЯЙЛО Я.О., студент
БАБЕНКО М.В., к.т.н., доцент
ЖУЛЬКОВСЬКИЙ О.О., к.т.н., доцент

Дніпровський державний технічний університет, м. Кам'янське

СТВОРЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОТРИМАННЯ НОТНОЇ ГРАМОТИ З МУЗИЧНИХ ФАЙЛІВ ФОРМАТУ MIDI

Вступ. Питання, пов'язані з програмуванням музики, «програмованою музикою», «музикою на основі розрахунків» тощо обговорюються досить тривалий час. Завдяки інтелектуалізації персональних комп'ютерів, наявності вбудованих систем аналітичних обчислень, великої кількості діалогових засобів роботи з табличними, текстовими, графічними, музичними об'єктами і т. д., а також у зв'язку з розвитком спеціального програмного забезпечення виникли реальні можливості синтезу композиції з теорією інформації, об'єднання музичних параметрів з акустичними за допомогою серійного комбінування.

Музичне програмування, яке передбачає детальне з'ясування нюансів уявлень про способи функціонування гармонії і тенденції її розвитку, зробило істотний внесок в розвиток сучасних уявлень про музичну гармонію. Дана робота присвячена вибору оптимального шляху перетворення музичних файлів у нотну грамоту завдяки аналізу музичних файлів формату MIDI за допомогою аспектів музикознавства, що допускають формалізований підхід з можливістю навчання гри на фортепіано будь-якої мелодії, що запущена за допомогою програми.

MIDI (англ. *Musical Instrument Digital Interface*, цифровий інтерфейс музичних інструментів) – стандарт передачі інформації між електронними музичними інструментами, розроблений 1983 року, що уможливорює комунікацію електромозичних інструментів, комп'ютера та іншого MIDI-сумісного обладнання, здійснювати з одного інструменту управління іншими.

MIDI не передає звукової інформації, натомість MIDI працює з «повідомленнями», такими як висота та динаміка взятої на інструменті ноти, контрольні сигнали для таких параметрів як гучність, панорама, сигнали відліку часу для синхронізації темпу тощо. Як електронний протокол MIDI відзначається надзвичайно широким поширенням.

Постановка задачі. Засобами мови програмування C# необхідно реалізувати алгоритм отримання нотної грамоти з музичних файлів формату MIDI, при цьому процес отримання зробити дуже швидким, що дозволить музикантам в режимі реального часу перетворити будь-яку мелодію в нотну грамоту з можливістю друку. З розвитком технологій музичного програмування в широкому і вузькому розумінні цього процесу виник і новий об'єктивний метод вивчення творчості – моделювання (відтворення або імітація) деяких сторін досліджуваних об'єктів або процесів. Таким чином, для можливості навчання початківців гри на фортепіано можна зімітувати процес гри на інструменті будь-якої мелодії, запущеної через програму, а для професіональних музикантів дуже зручним буде наявність отриманої з музичних файлів нотної грамоти з можливістю регулювання тональності, часової сигнатури, показника тривалості, швидкості відтворення, з можливістю транспозиції та модуляції, а також вибору інструментів для програвання музичного файлу [1].

Результати роботи. Підсистема MIDI, на відміну від звукової підсистеми, надає додаткам три (а не два) класи об'єктів: пристрій введення (In), пристрій виведення (Out) і буферизований потік (Stream). Відповідно, є три класи інтерфейсних функцій для обслуговування цих об'єктів. Імена функцій мають відповідні префікси – midiInStart, midiStreamOut тощо. Було виявлено, що найчастіше однойменні функції з різними префіксами розрізняються тільки видами об'єктів, до яких вони належать. Якщо ж сенс і поведінка функцій різняться – вони будуть описуватися і згадуватися окремо.

При відкриванні об'єкта (потіку) підсистема повертає його ідентифікатор або ключ (handle), за яким потім відбувається вся інша робота з об'єктом. Ключ потоку виведення має тип NMIDISTRM і в деяких функціях може використовуватися замість ключа пристрою виведення; в таких випадках потрібно явне приведення до типу NMIDIOUT [2].

MIDI-інтерфейс містить два рівня уніфікації – апаратний, що описує правила з'єднання пристроїв і передачі сигналів, і протокольний, що описує протокол взаємодії і види переданих повідомлень. В даній роботі розглянуто насамперед програмування передачі MIDI-повідомлень в середовищі Windows. Зовні підсистема MIDI дуже схожа на підсистему цифрового звуку Audio/Wave і містить практично той же набір функцій і структур.

Створене програмне забезпечення дозволяє перетворювати музичні файли у нотну грамоту з функцією відтворення на фортепіано будь-якої мелодії, яка програється у додатку. На рис.1 зображено, як у програмі виконується програвання мелодії з отриманням нотної грамоти та навчання тому, як зіграти її на фортепіано. У програмному додатку реалізовані спеціальні засоби мови програмування C# для роботи з нотами і, насамперед, зроблено так, аби кожна нота мелодії, що програється, підсвічувалася. Користувач має можливість обирати кольори для нот, що витягуються. Нотна грамота, отримана з мелодій, буде зберігатися у форматі PNG з можливістю друку.

У файлах MIDI час вимірюється у так званих «імпульсах» і для роботи з часовою складовою створено окремий клас, що буде використовуватися для перетворення тривалості імпульсів у тривалість нот. Ноти з подібним часом імпульсу (наприклад, 10 імпульсів) будуть звучати як акорди. Для полегшення читання нотної грамоти можна зображати ноти у вигляді акордів. В MIDI використовується рівномірною

темперований лад і 128 нот різної висоти (з номерами від 0 до 127). Частота нот задається за допомогою номера. Наприклад, нота з номером 60 – це «До» першої октави (частота 261 Гц).

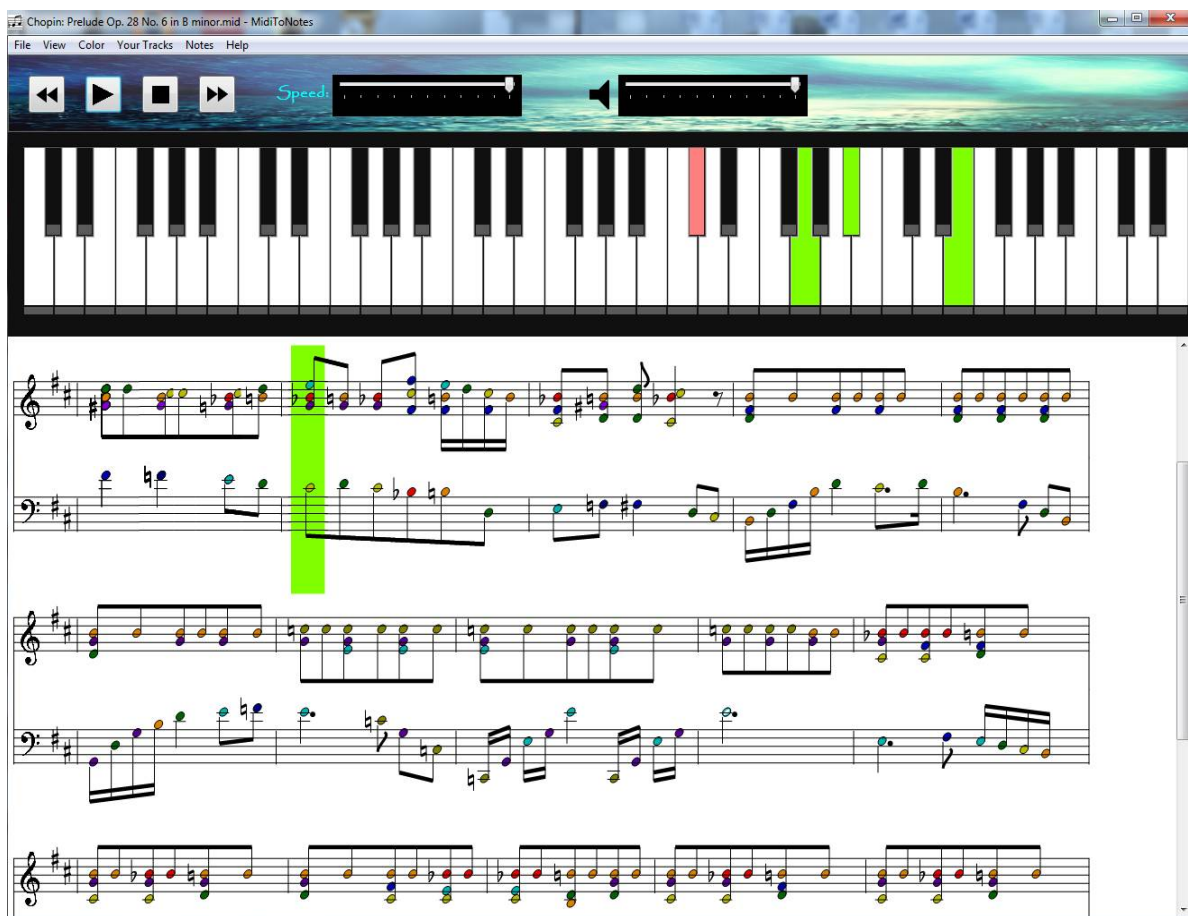


Рисунок 1 – Приклад роботи розробленого програмного забезпечення

Алгоритм отримання нотної грамоти виглядає наступним чином. Спочатку аналізується файл MIDI, отриманий на вході. Далі об'єкт класу, який отримує аналіз, після читання файлу буде містити повний список MIDI-подій, музичний розмір пісні, всі доріжки пісні, що складаються з нот, номер, час початку та тривалість кожної ноти. Визначаємо тривалість ноти, потім переходимо до іншої і т.д. В такий спосіб нотна грамота виглядає правильно, ноти та усі музичні символи розміщуються послідовно та збалансовано. Окремий клас відповідає за створення клавіатури фортепіано, за допомогою чого з'являється можливість демонстрації гри на інструменті тієї чи іншої мелодії, що програться.

Для реалізації алгоритму створено класи, які відповідають за музичні (нотні) символи (музичний розмір, ключові знаки, тактові смуги, нотні позначення і т.д.), для кожного типу символів – окремий клас, а також класи, які служать для роботи з форматом MIDI, наприклад, один з класів, що створює символ музичного розміру (дріб, що позначає ритмічну впорядкованість та величину такту в музиці) на початку кожного нотного стану. Біля зображення музичного розміру в нотній грамоті позначаються ключові знаки, за які відповідає окремий клас. Також розроблено клас, який являє собою вертикальні смуги, що обмежують такти. Часом початку тактової смуги є початок нового так-

ту. Крім того, створено клас музичних ключів, що відповідає за створення зображення скрипкового ключа або басового ключа, які можуть бути нормального чи маленького розмірів. Були розроблені класи для роботи з іншими музичними символами, у тому числі і з музичними паузами, що мають початковий час та тривалість. Окремий клас відповідає за можливі помилки при аналізі файлів. Конструктор приймає зсування файлу (в байтах), де сталася помилка, і рядок, що описує помилку. Для читання низькорівневих двійкових даних з файлу розроблено клас, який дає можливість прочитати наступний байт у файлі, читати рядки ASCII фіксованої довжини, читати задану кількість байтів у файлі. Один з класів містить доступні налаштування для модифікації нотної музики та звуку.

У додатку є можливість змінювати положення нот (вертикальне чи горизонтальне), збільшувати чи зменшувати нотні знаки. Деякі мелодії складаються з єдиної музичної доріжки, а інші з декількох доріжок (треків). Користувач має можливість обрати необхідний інструмент, за допомогою якого буде програватися доріжка (для кожної доріжки можна обрати свій інструмент).

MIDI є вираженим клавішно-орієнтованим протоколом, тому процес вилучення нот кодується двома простими повідомленнями – взяти ноту (Note On) і зняти ноту (Note Off). Виконавець при натисканні клавіші задає відразу три параметра: момент початку звучання, динаміку і висоту тону. Тривалість звуку визначається за моментом відпускання клавіші. При об'єднанні нот в доріжках створюється єдиний трек. Щоб об'єднати окремі треки розроблено нескладний алгоритм, подібний до сортування злиттям.

MIDI-повідомлення – це потік даних в реальному часі. Компоненти повідомлень в протоколі MIDI представлені байтами. Компонент, що описує тип повідомлення, називається статус-байтом, компонент, що уточнює повідомлення – байтом даних. Якщо уточнюючої інформації забагато, вона може бути представлена кількома байтами даних. Отже, кожне MIDI-повідомлення складається з одного статус-байта і, якщо необхідно, одного або декількох байтів даних. Для того, щоб пристрій міг безпомилково відрізнити статусний байт від байту даних, прийнято, що кожен старший біт статусного байту має значення «1», а кожний старший біт байту даних – «0». У байті даних решта 7 біт відведено для кодування значення того чи іншого параметра, що дозволяє закодувати 128 різних значень. У статусному байті наступні три біта кодують тип повідомлення, а останні 4 біта – один з 16 каналів повідомлення або тип системного повідомлення. Кількість байтів даних жорстко закріплено за кожним повідомленням. Для системних ексклюзивних повідомлень зроблено виняток – їх довжина жорстко не задається. Вона визначається спеціальним статус-байтом, який поміщається в кінець повідомлення.

У процесі розробки програмного забезпечення було виявлено, що для прийому коротких повідомлень у програмі досить передбачити обробник асинхронних подій типу DATA, який буде отримувати повідомлення від підсистеми MIDI кожен раз, коли на вхід інтерфейсу надійде чергове MIDI-повідомлення. Оскільки довжина коротких повідомлень не перевищує 3 байтів, вони передаються оброблювачу в числі параметрів функції або повідомлень, упакованих в змінну типу DWORD, і для їх прийому не потрібно виділення будь-яких буферів в пам'яті. Прийом повідомлень починається відразу ж після звернення до функції Start. Для виведення коротких повідомлень застосовується функція ShortMsg, в якій повідомлення передається цілком, упакованим в змінну типу DWORD [3].

Для прийому і виведення довгих повідомлень необхідно використовувати механізм буферизації за тією ж схемою, що і для запису/відтворення цифрового звуку.

Висновки. У сучасному електронному та комп'ютерному музичному інструментарії найбільш повно втілилися накопичені інформаційні технології в музиці. Все це вимагає, з одного боку, підготовки музикантів, які знаються на сучасних музично-комп'ютерних технологіях, а з іншого – підготовки фахівців технічного профілю, що мають основи загальної музичної освіти і володіють знаннями в області програмування звуку, звукосинтезу, аудіоінжиніринга, звукотембрального програмування, моделювання музично-творчих процесів і професійно володіють комп'ютерними програмами, фахівців, здатних займатися моделюванням як одним з перспективних методів об'єктивного дослідження музичної творчості.

Створене програмне забезпечення задовольняє нагальній потребі як початківців, так і професійних музикантів отримувати нотну інтерпретацію будь-яких мелодій у форматі MIDI в режимі реального часу з можливістю імітації гри цих мелодій на музичному інструменті.

ЛІТЕРАТУРА

1. Горбунова И.Б. Основы музыкального программирования: учеб. пос. / Горбунова И.Б., Заливадный М.С., Кибиткина Э.В. – СПб.: Изд-во РГПУ им. А.И.Герцена, 2012. – 195с.
2. Горбунова И.Б. Музыкально-компьютерные технологии: к проблеме моделирования процесса музыкального творчества: монография / Горбунова И.Б., Чибирев С.В. – СПб.: Изд-во РГПУ им. А. И. Герцена, 2012. – 160с.
3. Светлов М.Г. Системы искусственного интеллекта в интерактивной музыке, аудиовизуальных инсталляциях и перформансах / М.Г.Светлов // Современное музыкальное образование – 2010: междунар. науч.-практ. конф.: материалы / под общ. ред. И.Б.Горбуновой. – СПб., 2011. – С.131.

Надійшла до редколегії 29.01.2018.