



РОЗДІЛ 9

Ксилофон

Малюнок 9-1.



Важко повірити, що використання технологій для запису та відтворення музики почалося лише у 1878 році, коли Едісон запатентував фонограф. З тих пір ми зайшли так далеко - з музичними синтезаторами, компакт-дисками, семплами та реміксами, телефонами, які відтворюють музику, і навіть міжміськими джемами через Інтернет. У цьому розділі ви візьмете участь у цій традиції, створивши додаток Xylophone, який записуватиме та відтворюватиме музику.

Що ви побудуєте

За допомогою програми, показаної на *Рисунку 9-1* (створеної Ліз Луні з команди App Inventor), ви можете це зробити:

- екран.
- Грайте вісім різних нот, торкаючись кольорових кнопок на екрані.
- Натисніть кнопку Play, щоб відтворити ноти, які ви грали раніше.
- Натисніть кнопку скидання, щоб програма видалила всі ноти, які ви грали раніше, і ви змогли ввести нову пісню.

Що ви дізнаєтесь

У цьому підручнику розглядаються такі поняття:

- Використання одного компонента Sound для відтворення різних аудіофайли.



- Використання

компонента Clock для вимірювання та забезпечення затримок між діями.

Малюнок 9-2. Інтерфейс користувача програми Xylophone

- Вирішення питання про те, коли створювати процедуру.
- Створення процедури, яка викликає сама себе.
- Розширене використання списків, включаючи додавання елементів, доступ до них та очищення списку.

Початок роботи

Підключіться до веб-сайту App Inventor і створіть новий проект. Назвіть його "Xylophone", а також встановіть заголовок екрана "Xylophone". Підключіть свій додаток до пристрою або емулятора.

Проектування компонентів

Ця програма має 13 різних компонентів (8 з яких складають клавіатуру), які перелічено у *Таблиці 9-1*. Оскільки їх так багато, було б досить нудно створювати їх усі перед початком написання програми, тому ми розділимо додаток на функціональні частини і створюватимемо їх послідовно, переходячи між Конструктором і Редактором блоків, як ми це робили з додатком "Погоня за сонечком" у *Розділі 5*.

Таблиця 9-1. Всі компоненти для програми Xylophone

Тип компонента	Група палітри	Як ви його назвете	Мета
Кнопка	Інтерфейс користувача	Кнопка 1	Грай низьку клавішу до.
Кнопка	Інтерфейс користувача	Кнопка 2	Грай на клавіші "D".
Кнопка	Інтерфейс користувача	Кнопка 3	Грай на клавіші "E".
Кнопка	Інтерфейс користувача	Кнопка 4	Грай на клавіші "F".
Кнопка	Інтерфейс користувача	Кнопка 5	Грай на клавіші G.
Кнопка	Інтерфейс користувача	Кнопка 6	Зіграй "ля".
Кнопка	Інтерфейс користувача	Кнопка 7	Грай на клавіші "B".
Кнопка	Інтерфейс користувача	Кнопка 8	Грай на високій клавіші до.

Створення

Звук	Медіа	Звук 1	Грай по нотах.
Кнопка	Інтерфейс користувача	Кнопка відтворення	Увімкни пісню.
Кнопка	Інтерфейс користувача	ResetButton	Скинути пам'ять композиції.
Горизонтальне розташування	Макет	Горизонтальне розташування1	Розмістіть кнопки "Відтворення" та "Скидання" поруч один з одним.
Годинник	Інтерфейс користувача	Годинник 1	Відстежуйте затримки між нотатки.

Створення клавіатури

Наш користувацький інтерфейс включатиме восьминотну клавіатуру для пентатонічної (семиночної) мажорної гами від низького до високого до. У цьому розділі ми створимо цю музичну клавіатуру.

СТВОРЕННЯ ПЕРШИХ КНОПОК НОТАТОК

Почніть зі створення перших двох клавіш ксилофона, які ми реалізуємо як кнопки.

- З категорії Інтерфейс користувача перетягніть кнопку на екран. Назвіть її Button1. Ми хочемо, щоб це була довга пурпурна смужка, як на ксилофоні, тому налаштуйте її властивості наступним чином:
 - Змініть властивість BackgroundColor на Magenta.
 - Змініть властивість Text на "C".
 - Встановіть властивість Width (Ширина) на "Fill parent" (Заливка батька) так, щоб вона охоплювала весь екран.
 - Встановіть властивість "Висота" на 40 пікселів.
- Повторіть те саме для другої кнопки з назвою Button2, розташувавши її під кнопкою Button1. Використовуйте ті самі значення властивостей Width і Height, але встановіть для них значення BackgroundColor - червоний, а для Text - "D".

(Пізніше ми повторимо крок 2 для ще шести кнопок нот).

Вигляд у Конструкторі компонентів має виглядати приблизно так, як показано на *рисунку 9-2*.

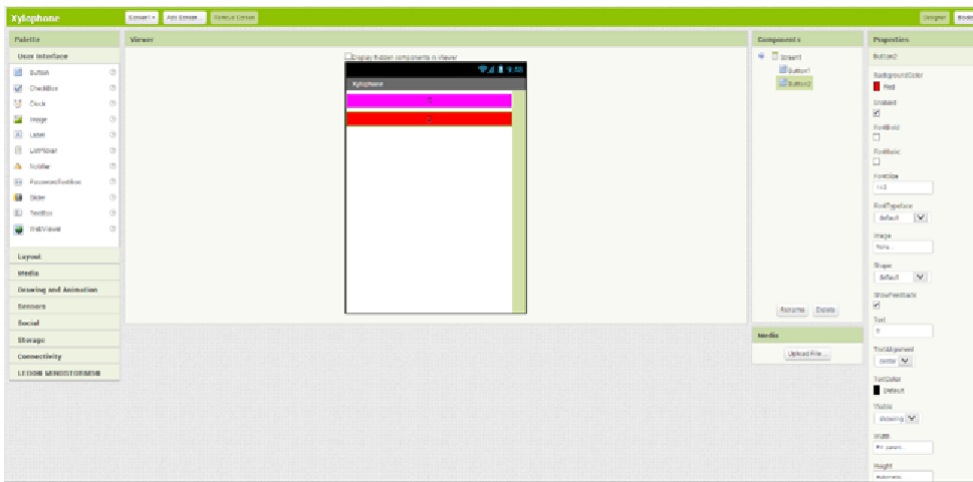


Рисунок 9-3. Розміщення кнопок для створення клавіатури

Дисплей на вашому телефоні має виглядати аналогічно, хоча між двома кольоровими кнопками не буде порожнього простору.

ДОДАВАННЯ ЗВУКОВОЇ СКЛАДОВОЇ

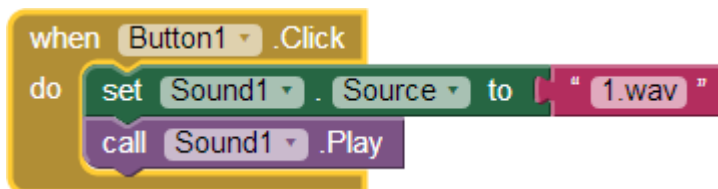
Ми не можемо мати ксилофон без звуків, тому перетягніть компонент Sound, залишивши його ім'я Sound1. Змініть властивість MinimumInterval зі значення за замовчуванням 500 мілісекунд на 0. Це дозволить нам відтворювати звук так часто, як ми хочемо, замість того, щоб чекати півсекунди (500 мілісекунд) між відтвореннями. Не встановлюйте його властивість Source, яку ми встановимо у редакторі блоків.

Завантажте звукові файли: <http://appinventor.org/bookFiles/Xylophone/1.wav> та <http://appinventor.org/bookFiles/Xylophone/2.wav>. На відміну від попередніх розділів, де можна було змінювати назви медіафайлів, тут важливо використовувати саме ці назви з причин, які незабаром стануть зрозумілими. Решту шість звукових файлів ви можете завантажити пізніше.

ПІДКЛЮЧЕННЯ ЗВУКІВ ДО КНОПОК

Поведінка, яку нам потрібно запрограмувати, полягає у відтворенні звукового файлу при натисканні відповідної кнопки. Зокрема, якщо натиснуто кнопку Button1, ми хочемо відтворити файл 1.wav; якщо натиснуто кнопку Button2, ми хочемо відтворити файл 2.wav; і так далі. Ми можемо налаштувати це у редакторі блоків, як показано на малюнку 9-3, виконавши такі дії:

1. Перетягніть блок Button1.Click зі скриньки Button1.
2. З шухляди Sound1 перетягніть встановлений блок Sound1.Source, розмістивши його на панелі
Кнопка 1. Блок клацання.
3. Введіть "text", щоб створити текстовий блок. (Це швидше, ніж перейти на вкладку "Вбудовані", а потім на панель "Текст", хоча цей спосіб теж підійде). Встановіть для нього значення "1.wav" і помістіть його у блок Sound1.Source.
4. Додайте блок Sound1.Play.



Малюнок 9-4. Відтворення звуку при натисканні кнопки

Ми могли б зробити те саме для Button2, як показано на *рисунок 9-4* (просто змінивши текстове значення), але код буде жахливо повторюватися.



Малюнок 9-5. Додавання нових звуків

Повторюваний код - це добра ознака того, що вам слід створити процедуру, що ви вже робили у грі MoleMash у Розділі 3 та у грі Ladybug Chase у Розділі 5. Зокрема, ми створимо процедуру, яка отримує число як параметр, встановлює джерело Sound1 на відповідний файл і відтворює звук. Це ще один приклад *рефакторингу* - покращення реалізації програми без зміни її поведінки, поняття, яке було введено у підручнику з MoleMash. Ми можемо використати блок приєднання Text drawer для об'єднання числа (наприклад, 1) і тексту ".wav", щоб створити відповідне ім'я файлу (наприклад, "1.wav"). Нижче наведено кроки для створення потрібної нам процедури:

1. На вкладці Вбудовані перейдіть до скриньки Процедури і перетягніть блок to procedure do. (Якщо не вказано інше, вам слід вибрати версію з "do", а не "result").
2. Додайте параметр, натиснувши на маленьку синю іконку на блоці to procedure do, перетягнувши на вхід і змінивши його назву з "x" на "number". Можливо, ви захочете переглянути Рисунок 5-6 з Розділу 5.
3. Клацніть на назві процедури, яка за замовчуванням є "procedure", і встановіть для неї значення "PlayNote".
4. Перетягніть блок Sound1.Source з Button1.Click у PlayNote праворуч від слова "do". Перенесіть блок Sound1.Play також у PlayNote.
5. Перетягніть блок 1.wav у кошик.
6. Перетягніть блок з'єднання зі скриньки Text до гнізда Sound1.Source.
7. Введіть "number" і перемістіть його у верхній сокет блоку з'єднання (якщо його там ще немає).
8. Перетягніть текстовий блок зі скриньки "Текст" у друге гніздо з'єднання блок.
9. Змініть текстове значення на ".wav". (Не забудьте не ставити лапки).
10. Зі скриньки Процедури перетягніть блок виклику PlayNote і помістіть його у порожнє тіло Button1.Click.
11. Введіть "1" і введіть його в гніздо "номер".

Тепер при натисканні кнопки `Button1` буде викликано процедуру `PlayNote`, параметр `number` якої має значення 1. Вона має встановити `Sound1.Source` у "1.wav" і відтворити звук.

Створіть аналогічний блок `Button2.Click` з викликом `PlayNote` з параметром 2 (ви можете скопіювати існуючий блок виклику `PlayNote` і перемістити його в тіло `Button2.Click`, обов'язково змінивши параметр). Ваша програма повинна мати вигляд, як показано на *рисунку 9-5*.

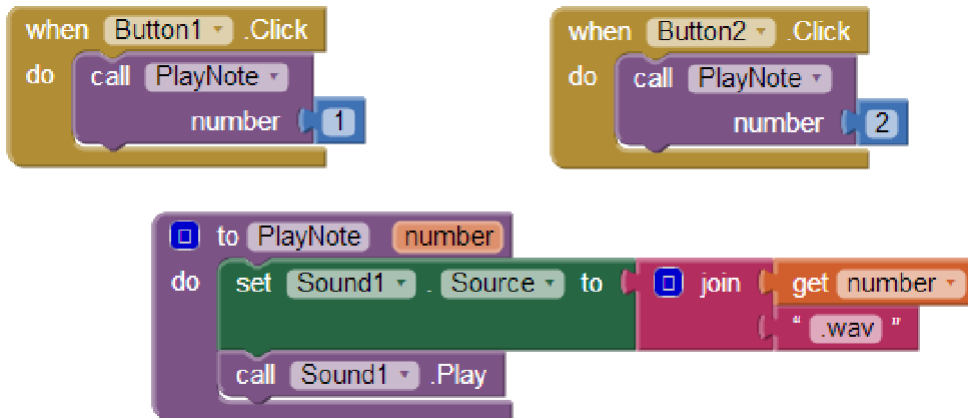
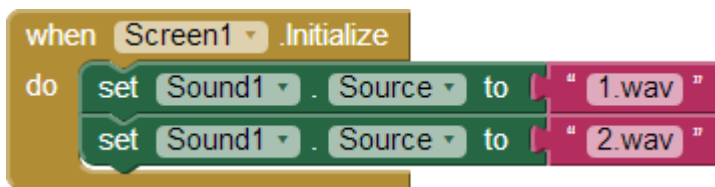


Рисунок 9-6. Створення процедури для відтворення ноти

ІНСТРУКЦІЯ ДЛЯ АНДРОЇДА ПО ЗАВАНТАЖЕННЮ ЗВУКІВ

Якщо ви спробували попередні дзвінки у `PlayNote`, ви могли бути розчаровані, не почувши очікуваного звуку або зіткнувшись з помилкою чи несподіваною затримкою. Це тому, що Android має завантажувати звуки під час виконання, що спричиняє деяку затримку, перш ніж вони будуть відтворені. Ця проблема не виникала раніше, тому що імена файлів, розміщені у властивості `Source` компонента `Sound` у Дизайнері, автоматично завантажуються під час запуску програми. Оскільки ми не встановлюємо `Sound1.Source` до запуску програми, процес ініціалізації не відбувається. Ми повинні явно завантажити звуки під час запуску програми, як показано на *рисунку 9-6*.



Малюнок 9-7. Звуки завантаження під час запуску програми



Протестуйте додаток Торкніться кнопок і перевірте, чи відтворюються ноти без затримки. (Якщо ви нічого не чуєте, переконайтеся, що на телефоні не вимкнено звук).

ВИКОНАННЯ РЕШТИ ЗАУВАЖЕНЬ

Тепер, коли перші дві кнопки і ноти реалізовані і працюють, додайте решту шість нот, повернувшись до Дизайнера і завантаживши звукові файли:

- <http://appinventor.org/bookFiles/Xylophone/3.wav>
- <http://appinventor.org/bookFiles/Xylophone/4.wav>
- <http://appinventor.org/bookFiles/Xylophone/5.wav>
- <http://appinventor.org/bookFiles/Xylophone/6.wav>
- <http://appinventor.org/bookFiles/Xylophone//.wav>
- <http://appinventor.org/bookFiles/Xylophone/8.wav>

Потім створіть шість нових кнопок, виконавши ті самі кроки, що й для попередніх двох, але налаштувавши їхні властивості Text і BackgroundColor наступним чином:

- Кнопка3 ("E", рожева)
- Button4 ("F", помаранчевий)
- Кнопка5 ("G", жовта)
- Кнопка6 ("A", зелена)
- Button7 ("B", блакитний)
- Кнопка8 ("C", синя)

Ви також можете змінити властивість TextColor кнопки Button8 на White, як показано на рисунку

Малюнок 9-7, щоб він був більш читабельним.

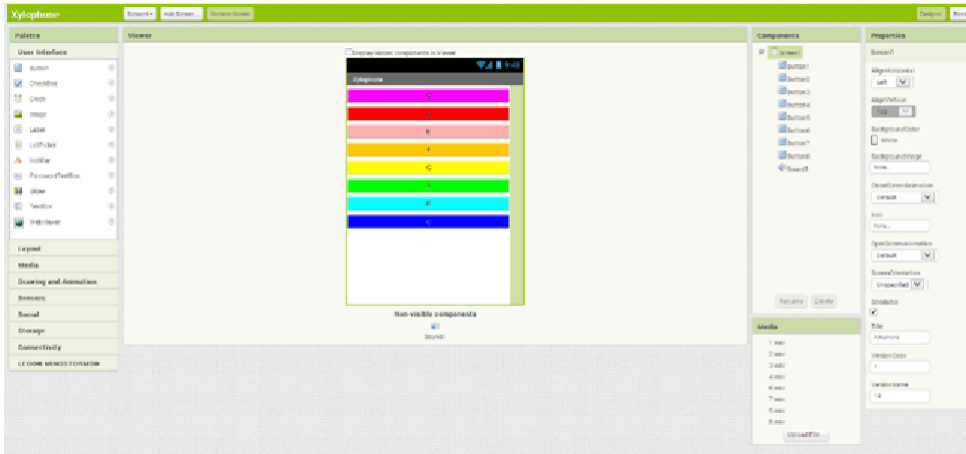
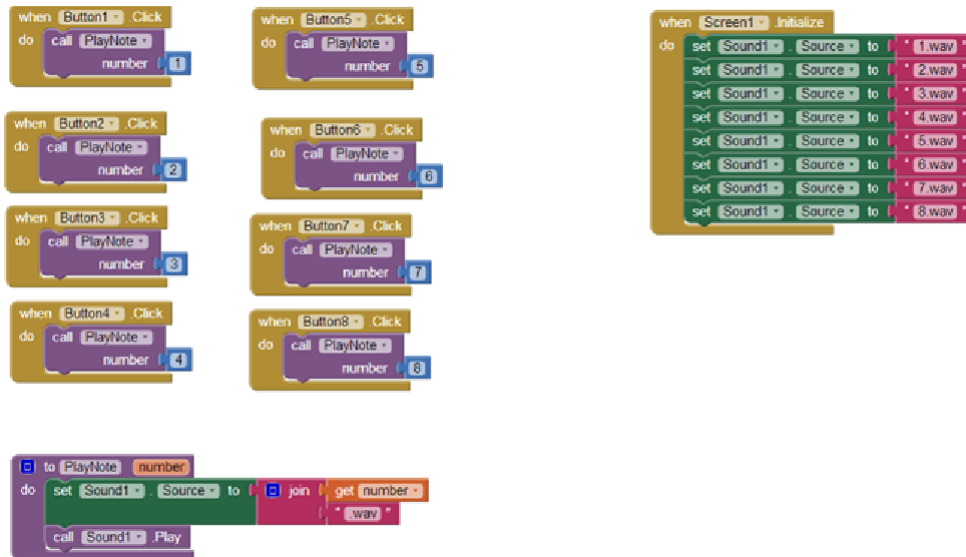


Рисунок 9-8. Додавання решти кнопок та звуків у Дизайнер компонентів

Поверніться до редактора блоків і створіть блоки Click для кожної з нових кнопок з відповідними викликами PlayNote. Так само додайте кожен новий звуковий файл до Screen.Initialize, як показано на *рисунку 9-8*.



Малюнок 9-9. Програмування подій натискання кнопки на відповідність усім клавішам клавіатури



Протестуйте свій додаток Тепер у вас повинні бути всі кнопки, і кожна з них буде відтворювати окрему ноту, коли ви натискаєте на неї.

Запис і відтворення нот

Грати ноти натисканням кнопок - це весело, але мати можливість записувати і відтворювати пісні - ще краще. Щоб реалізувати відтворення, нам потрібно вести запис зіграних нот. Окрім запам'ятовування висоти звуку (звукових файлів), які були зіграні, ми також повинні записувати час між нотами, інакше ми не зможемо відрізнити дві ноти, зіграні швидко, від двох, зіграних з 10-секундною перервою між ними.

Наш додаток буде підтримувати два списки, в кожному з яких буде по одному запису для кожної ноти, що була зіграна:

- нотатки, які будуть містити назви звукових файлів у тому порядку, в якому вони були відтворені.
- times, які будуть фіксувати моменти часу, в які були зіграні ноти.



Примітка *Перед тим, як продовжити, ви, можливо, захочете переглянути списки, про які йдеться у Вікторині для президентів у Розділі 8 та Розділі 19.*

Ми можемо отримати інформацію про час з компонента Clock, який ми також використовуємо для правильного визначення часу відтворення нот.

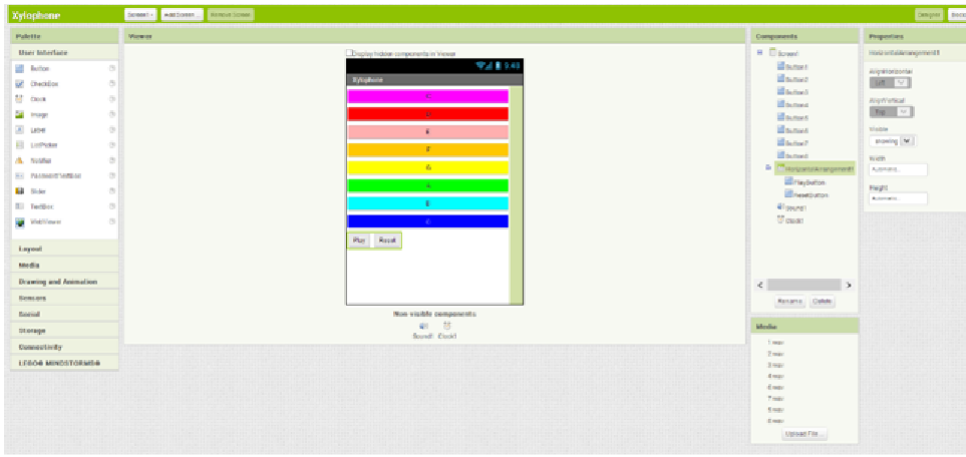
ДОДАВАННЯ КОМПОНЕНТІВ

У Дизайнері вам потрібно буде додати компонент Годинник і кнопки Відтворення та Скидання, які ви розмістите в горизонтальному розташуванні (HorizontalArrangement):

1. Зі скриньки "Датчики" перетягніть компонент "Годинник". Він з'явиться у розділі "Невидимі компоненти". Зніміть позначку з його властивості TimerEnabled, оскільки ми не хочемо, щоб його таймер спрацьовував, доки ми не скажемо йому про це під час відтворення.
2. Перейдіть до панелі "Макет" і перетягніть компонент "Горизонтальне розташування" під наявною кнопкою. Встановіть для його властивості Width значення "Заповнення батьківського елемента".
3. Перетягніть кнопку зі скриньки "Інтерфейс користувача". Перейменуйте її на "PlayButton" і встановіть для її властивості Text значення "Play".
4. Перетягніть іншу кнопку, розмістивши її праворуч від кнопки PlayButton. Перейменуйте нову кнопку

Кнопка "ResetButton" і встановіть її властивість Text у значення "Reset".

Вигляд Конструктора має виглядати, як показано на *рисунку 9-9*.



Малюнок 9-10. Додавання компонентів для запису та відтворення звуків

ЗАПИС НОТАТОК І ЧАСУ

Тепер нам потрібно додати правильну поведінку в редакторі блоків. Нам потрібно буде вести списки нотаток і часу і додавати до них щоразу, коли користувач натискає кнопку.

1. Створіть нову змінну, перейшовши до панелі Змінні і перетягнувши ініціалізувати глобальний блок з шухляди визначень.
2. Змініть назву змінної на "notes".
3. Відкрийте скриньку Списки і перетягніть блок створення порожнього списку, розмістивши його у гнізді ініціалізації глобального блоку to.

Це визначає нову змінну з назвою "notes" як порожній список. Повторіть ці кроки для іншої змінної, яку слід назвати "times". Ці нові блоки мають виглядати так, як показано на *рисунку 9-10*.

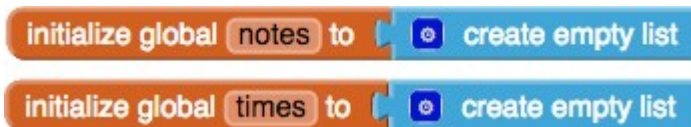
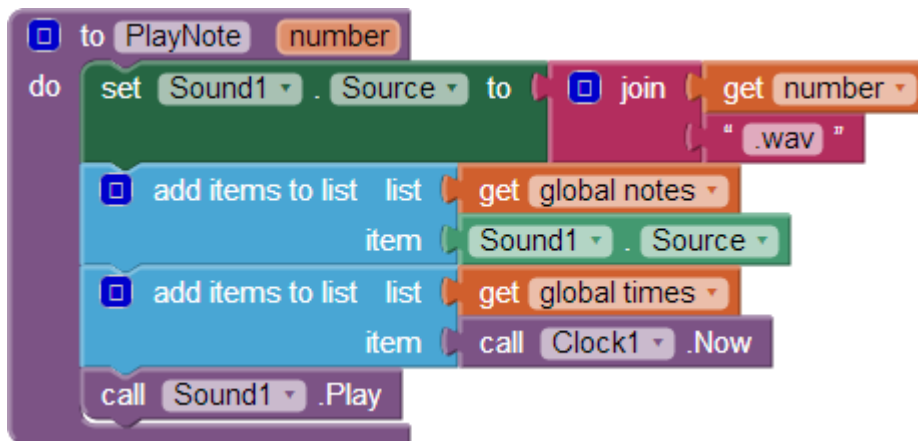


Рисунок 9-11. Ініціалізуйте дві змінні для зберігання нотаток та інформації про час

Як працюють блоки

Кожного разу, коли звучить нота, нам потрібно зберігати як назву звукового файлу (у списку нот), так і момент часу, коли вона була зіграна (у списку часів). Щоб записати

Для отримання поточного моменту часу скористаємося блоком `Clock1.Now`, який повертає поточний момент часу (наприклад, 12 березня 2011 року, 8:33:14) з точністю до мілісекунди. Ці значення, отримані за допомогою блоків `Sound1.Source` і `Clock1.Now`, слід додати до списків `notes` і `times` відповідно, як показано на *рисунку 9-11*.

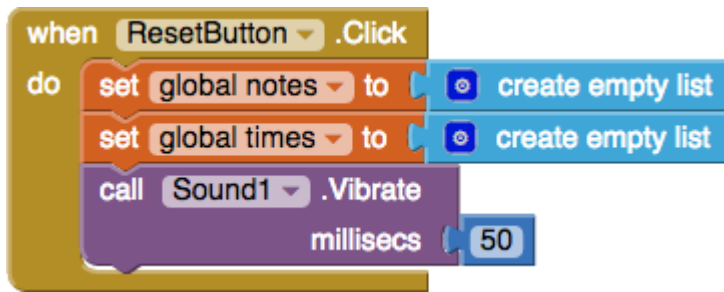


Малюнок 9-12. Додавання відтворених звуків до списку

Наприклад, якщо ви зіграєте "Row, Row, Row Your Boat" [C C C D E], у вашому списку буде п'ять записів, які можуть виглядати наступним чином:

- ноти: 1.wav, 1.wav, 1.wav, 2.wav, 3.wav
- час [дати пропущено]: 12:00:01, 12:00:02, 12:00:03, 12:00:03.5, 12:00:04

Коли користувач натискає кнопку Reset, ми хочемо, щоб обидва списки повернулися до початкового, порожнього стану. Оскільки користувач не побачить жодних змін, добре було б додати невеликий блок Sound1.Vibrate, щоб показати, що натискання клавіші було зареєстровано. На *рисунку 9-12* показано блоки для такої поведінки.

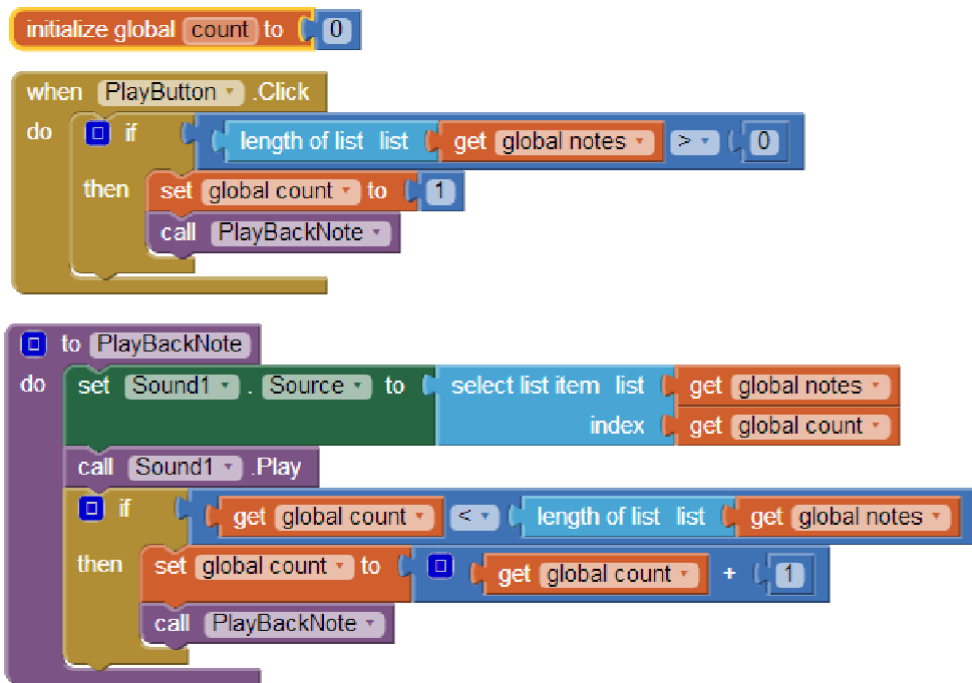


Малюнок 9-13. Надання зворотного зв'язку, коли користувач перезавантажує додаток

ВІДТВОРЕННЯ НОТ

У якості уявного експерименту, давайте спочатку подивимося, як реалізувати відтворення нот, не турбуючись про хронометраж. Ми могли б (але не будемо) зробити це, створивши такі блоки, як показано на *рисунку 9-13*:

- Змінна кількість, щоб відстежувати, на якій ноті ми перебуваємо.
- Нова процедура PlayBackNote, яка відтворює цю ноту і переходить до наступної.
- Код, що запускається при натисканні кнопки PlayButton, який встановлює лічильник на 1 і викликає PlayBackNote, якщо немає збережених нот.



Малюнок 9-14. Відтворення записаних нот

Як працюють блоки

Можливо, ви вперше бачите процедуру, яка викликає сама себе. Хоча на перший погляд це може здатися несправжнім, насправді це важлива і потужна концепція комп'ютерних наук, яка називається *рекурсія*.

Щоб краще зрозуміти, як працює рекурсія, давайте розглянемо, що станеться, якщо користувач зіграє/запише три ноти (1.wav, 3.wav і 6.wav), а потім натисне кнопку Play. Спочатку запускається PlayButton.Click. Оскільки довжина нот у списку дорівнює 3, що більше за 0, значення count дорівнює 1, і викликається PlayBackNote:

1. При першому виклику PlayBackNote, count = 1:
 - Джерело звуку Sound1.Source встановлено на перший елемент у примітках, тобто 1.wav.
 - Звук1.Play викликається, відтворюючи цю ноту.
 - Оскільки кількість нот (1) менша за довжину нот (3), кількість збільшується до 2, і знову викликається PlayBackNote.
2. При другому виклику PlayBackNote, count = 2:
 - Джерело звуку Sound1.Source встановлено на другий пункт у примітках, тобто 3.wav.
 - Звук1.Play викликається, відтворюючи цю ноту.

- Оскільки кількість нот (2) менша за довжину нот (3), кількість збільшується до 3, і знову викликається `PlayBackNote`.
3. При третьому виклику `PlayBackNote`, `count = 3`:
- `Sound1.Source` встановлено на третій пункт у примітках, тобто `6.wav`.
 - `Звук1.Play` викликається, відтворюючи цю ноту.
 - Оскільки кількість (3) не менша за довжину нот (3), більше нічого не відбувається, і відтворення завершується.

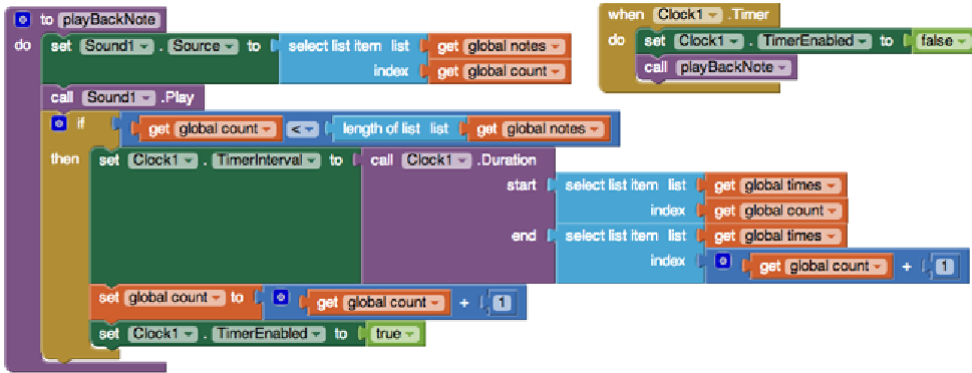


Зауваження Хоча рекурсія є потужною, вона також може бути небезпечною. У якості уявного експерименту запитайте себе, що сталося б, якби програміст забув вставити у `PlayBackNote` блоки, які збільшують лічильник.

Хоча рекурсія коректна, у попередньому прикладі є інша проблема: між викликом `Sound1.Play` і наступним викликом майже не проходить часу, тому кожна нота переривається наступною нотою, крім останньої. Жодна нота (окрім останньої) не може завершитися до того, як джерело `Sound1` буде змінено і знову викликано `Sound1.Play`. Щоб досягти коректної поведінки, нам потрібно реалізувати затримку між викликами `PlayBackNote`.

ВІДТВОРЕННЯ НОТ З НАЛЕЖНИМИ ЗАТРИМКАМИ

Ми реалізуємо затримку, встановивши таймер на годиннику на проміжок часу між поточною нотою і наступною нотою. Наприклад, якщо наступна нота буде зіграна через 3 000 мілісекунд (3 секунди) після поточної, ми встановимо `Clock1.TimerInterval` рівним 3 000, після чого знову викличемо `PlayBackNote`. Внесіть зміни, показані на *рисунку 9-14*, в тіло блоку `if` у `PlayBackNote`, а також створіть і заповніть обробник події `Clock1.Timer`, який визначає, що має статися, коли спрацює таймер.



Зображення 9-15. Додавання затримок між нотами

Як працюють блоки

Припустимо, що обидва списки мають наступний зміст:

- ноти: 1.wav, 3.wav, 6.wav
- час: 12:00:00, 12:00:01, 12:00:04

Як показано на *рисунок 9-14*, PlayButton.Click встановлює лічильник на 1 і викликає PlayBackNote.

1. При першому виклику PlayBackNote, count = 1:
 - Sound1.Source встановлено на перший елемент у примітках, тобто "1.wav".
 - Звук1.Play викликається, відтворюючи цю ноту.
 - Оскільки кількість нот (1) менша за довжину нот (3), Clock1.TimerInterval встановлюється рівним кількості часу між першим (12:00:00) та другим елементами у часах (12:00:01): 1 сек. відлік збільшується до 2. Clock1.Timer дорівнює увімкнена і починає зворотній відлік.

Протягом 1 секунди більше нічого не відбувається, після чого запускається Clock1.Timer, який тимчасово вимикає таймер і викликає PlayBackNote.

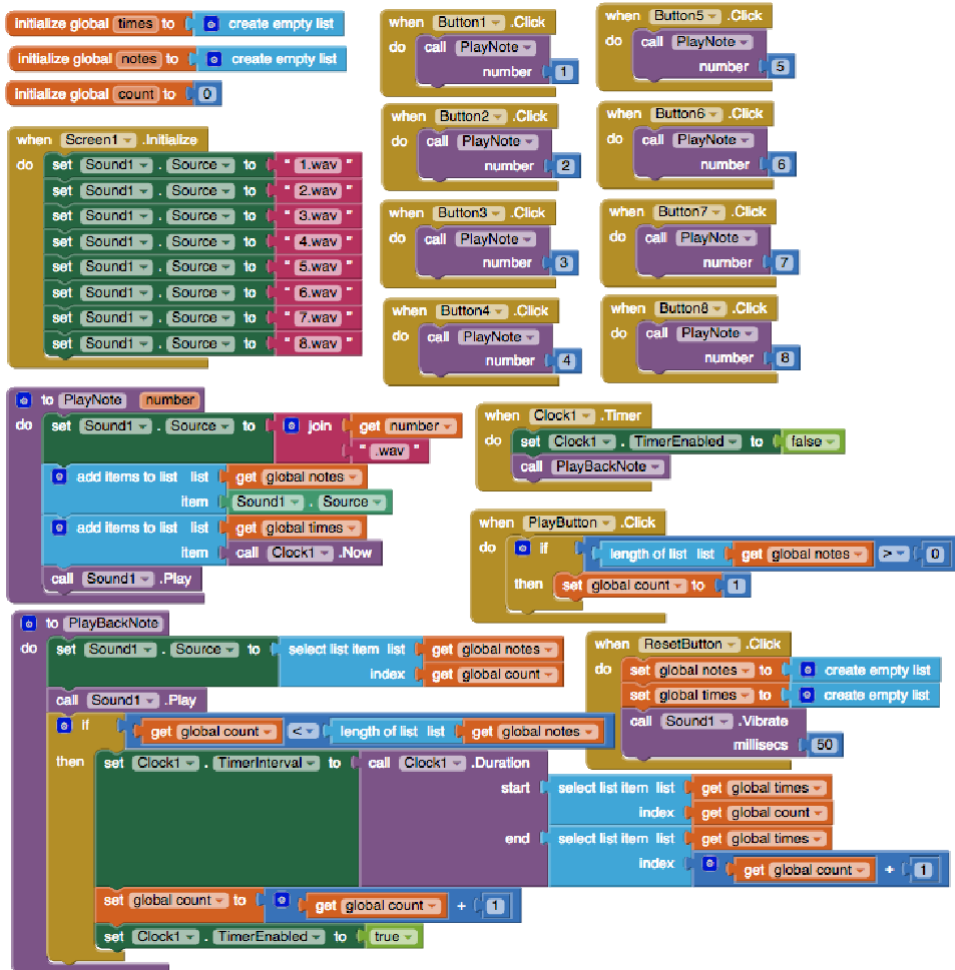
2. При другому виклику PlayBackNote, count = 2:
 - Sound1.Source встановлено на другий пункт у примітках, а саме "3.wav".
 - Звук1.Play викликається, відтворюючи цю ноту.
 - Оскільки кількість (2) менша за довжину нот (3), Clock1.TimerInterval встановлюється на час між другим (12:00:01) і третім елементами у часі (12:00:04): 3 секунди. відлік збільшується до 3. Clock1.Timer увімкнено і починає зворотній відлік.

Протягом 3 секунд більше нічого не відбувається, після чого запускається `Clock1.Timer`, який тимчасово вимикає таймер і викликає `PlayBackNote`.

3. При третьому виклику `PlayBackNote`, `count = 3`:
 - `Sound1.Source` встановлено на третій пункт у примітках, тобто "6.wav".
 - `Звук1.Play` викликається, відтворюючи цю ноту.
 - Оскільки кількість (3) *не* менша за довжину нот (3), більше нічого не відбувається. Відтворення завершено.

Повний додаток: Ксилофон

На *рисунку 9-15* показано остаточну конфігурацію блоків для програми `Xylophone`.



Малюнок 9-16. Блоки для ксилофона

Варіації

Ось кілька альтернативних сценаріїв для вивчення:

- Наразі ніщо не заважає користувачеві натиснути кнопку ResetButton під час відтворення, що призведе до аварійного завершення роботи програми. (Чи можете ви з'ясувати чому?) Змініть PlayButton.Click так, щоб він вимикав ResetButton. Щоб знову увімкнути її після завершення відтворення пісні, змініть блок if у PlayButton.Click на блок if else і знову увімкніть ResetButton у частині else.
- Аналогічно, користувач може натиснути кнопку PlayButton, коли пісня вже грає. (Чи можете ви зрозуміти, що станеться?) Зробіть так,

щоб `PlayButton.Click`

вимикає кнопку `PlayButton` і змінює її текст на "Playing...". Ви можете знову увімкнути її і скинути текст у блоці `ifelse`, як описано у попередньому пункті.

- Додайте кнопку з назвою пісні, наприклад, "Für Elise". Якщо користувач натисне її, заповнить списки нот і часу відповідними значеннями, встановить кількість на 1 і викличе `PlayBackNote`. Для встановлення відповідного часу вам стане у пригоді блок `Clock1.MakeInstantFromMillis`.
- Якщо користувач натискає ноту, відходить і робить щось інше, а потім повертається через кілька годин і натискає додаткову ноту, ноти будуть частиною однієї пісні, що, ймовірно, не те, що мав на увазі користувач. Удоскональте програму, 1) зупинивши запис через певний розумний проміжок часу, наприклад, хвилину; або, 2) встановивши обмеження на кількість часу, що використовується для `Clock1.TimerInterval`, за допомогою блоку `max` з математичної панелі `Math`.
- Візуально вкажіть, яка нота грає, змінивши вигляд кнопки - наприклад, змінивши її текст, колір тла або колір переднього плану.

Підсумок

Ось деякі з ідей, які ми розглянули в цьому підручнику:

- Ви можете відтворювати різні аудіофайли з одного звукового компонента, змінивши його властивість `Source`. Це дозволило нам мати один компонент `Sound` замість восьми. Тільки не забудьте завантажити звуки під час ініціалізації, щоб уникнути затримок (Малюнок 9-6).
- Списки можуть надавати програмі пам'ять, коли запис дій користувача зберігається у списку, а потім може бути витягнутий і перероблений. Ми використали цю функціональність для запису та відтворення пісні.
- Для визначення поточного часу можна використовувати компонент `Clock`. Віднімання двох значень часу дає нам кількість часу між двома подіями.
- Ви можете встановити властивість `TimerInterval` для `Clock` у програмі, наприклад, як ми встановили її на тривалість часу між початком двох нот.
- Процедура не тільки може, але й іноді бажано, щоб вона сама себе викликала. Це потужний прийом, який називається *рекурсією*. При

написанні рекурсивної процедури переконайтеся, що існує базовий випадок, в якому процедура завершується, а не викликає сама себе, інакше програма буде нескінченно циклічно повторюватися.