Taller “Steganos with MIDI”.

Santiago Hernández Facio Lince 201922432

El script: tallermidi.py, hace uso de la librería mido para manipular los archivos tipo midi. Leyendo un poco la documentación de la librería: <https://mido.readthedocs.io/en/stable/> , me di cuenta que permite incluir mensajes de tipo Sysex en los archivos midi.

Estos mensajes permiten agregar data al archivo, sin que este afecte el sonido del mismo. Este campo data, debe ser una lista de Integers que representan los valores ASCII de las letras del mensaje. Las únicas condiciones es que el valor de los números debe estar de 0 a 127 y la lista no puede tener más de un millón de elementos, acá la documentación de este tipo de mensajes: <https://mido.readthedocs.io/en/stable/messages/index.html>

Estas 2 restricciones presentaron diversos problemas. El primero fue con los caracteres especiales del español, como las tildes: á í é o también la ñ, ya que los valores ASCII de estos caracteres era mayor a 127, esto se solucionó añadiendo el valor 19 a la lista, para identificar que posterior a este 19 viene un carácter especial (se utilizó el valor 19 porque corresponde a un valor ASCII que casi nunca es utilizado en los archivos .txt), luego de agregar el carácter 19, al valor ASCII del carácter especial se le resta 127 y se agrega a la lista. Para decodificarlo, al momento de encontrar un 19 este es ignorado y al siguiente carácter que aparezca se le suma 127 para obtener su valor ASCII real y decodificarlo correctamente.

El segundo problema es que cada mensaje tiene un límite de un millón de números, por ende, al querer esconder toda una obra como el quijote, se debía calcular el tamaño total de la obra y crear los suficientes mensajes Sysex para cubrir toda la obra y que ningún mensaje pasara el límite de un millón.

Con estos problemas solucionados el script funciona correctamente.

**Funcionamiento:**

El script funciona bajo la línea de comando y el uso de flags, presentado las siguientes:

|  |  |
| --- | --- |
| -a | Este se refiere a la acción a realizar, “encode” para esconder el mensaje y “decode” para recuperar el mensaje y eliminarlo del archivo midi. |
| -o | Especificar la ruta del archivo midi original. |
| -d | Especificar el nombre del archivo resultante. Esta flag sirve para generar un archivo midi adicional con el mensaje escondido, sirve para no modificar el archivo midi original. Si el flag no es especificado, el mensaje se esconderá en el archivo midi original. |
| -m | Mensaje a esconder. |
| -f | Si se desea esconder el contenido de un archivo txt, se debe usar el flag -f con el nombre del archivo y NO utilizar el flag -m |

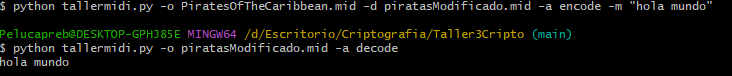
Ejemplo: Esconder el mensaje “hola mundo” en un archivo midi original llamado: PiratesOfTheCaribean.mid y guardar el mensaje en una copia del archivo llamado: piratasModificado.mid el cual no presenta ninguna alteración en el audio del archivo con respecto a al archivo original.

Comando: Python tallermidi.py -o PiratesOfTheCaribbean.mid -d piratasModificado.mid -a encode -m "hola mundo"

Para revelar el mensaje se debe poner el flag -o con el nombre del archivo midi que contiene el mensaje y el flag -a decode

Comando: python tallermidi.py -o piratasModificado.mid -a decode

Obteniendo el mensaje por consola:



Si se desea encriptar un archivo txt en vez de un mensaje escrito en consola el comando es el siguiente:

python tallermidi.py -o PiratesOfTheCaribbean.mid -d piratasModificado.mid -a encode -f elQuijote.txt

Como se mencionó anteriormente, se solucionaron los problemas de longitud y de archivos especiales, por lo que el código puede esconder archivos de cualquier longitud, como se puede ver en el ejemplo, escondiendo varios capítulos de el Quijote y no afecta para nada el audio del archivo, y también puede esconder tanto mensajes tanto en Ingles como en Español.

Para decodificar el quijote se usa el mismo comando anterior:

Comando: python tallermidi.py -o piratasModificado.mid -a decode