Fita 1

Configuració de l'entorn i proves

Màster Interuniversitari en Seguretat de les Tecnologies de la Informació i Comunicació

Cristian Requena Barreda

DECISIONS DE DISSENY

Tal com es va establir al document de fonaments, per tal de fer una primera prova de concepte de la captura d'àudio, s'ha començat a desenvolupar un petit programari per capturar àudio mitjançant la biblioteca OpenAL.

Aquesta biblioteca de baix nivell disposa de múltiples *wrappers* que fan possible el seu ús a diverses plataformes/tecnologies de desenvolupament. Com que un dels objectius era desenvolupar un programari tant d'alt nivell com fos possible, s'ha fet una primera aproximació mitjançant un petit aplicatiu Java.

S'han provat dues biblioteques d'enllaç amb OpenAL:

- JNAL: http://urish.org/openal/.
  Biblioteca poc documentada que tot i semblar suportar completament OpenAL, només disposa de la funcionalitat de reproducció de so, pel que no serveix per capturar àudio.
- JOAL: http://jogamp.org/.
  Tot i suportar la captura de so, s'ha vist que el seu ús és complex, pel que abans de dedicar-hi un elevat esforç, s'ha optat per cercar altres alternatives.

La màquina virtual de Java disposa d'un conjunt de classes (javax.sound) que permeten un accés d'alt nivell pel que fa a l'entrada i sortida d'àudio. El seu ús és senzill, i permet comprovar com es computa el so.

S'ha decidit realitzar aquesta avaluació mitjançant el mètode d'entrada/sortida estàndard de Java, ja que només variarà la metodologia de captura respecte la proposta d'ús d'OpenAL. En efecte, l'àudio computat serà equivalent emprant ambdues biblioteques.

DESENVOLUPAMENT

Una vegada establertes les decisions de disseny prèvies, es detallarà la implementació d'una captura d'àudio simple.

- Cerca del dispositiu d'entrada.
  Actualment els computadors domèstics disposen de múltiples entrades i sortides d'àudio (connectors frontals i posteriors, interfícies HDMI, etc.).
  Degut a aquesta circumstància, és necessari identificar el dispositiu mitjançant el qual es capturarà l'àudio, pel que cal cercar-lo d'entre els disponibles.
  En aquest cas, s'accedeix al dispositiu mitjançant el seu nom:

```
Mixer mixer = null;

for (Info info : AudioSystem.getMixerInfo()) {
        if (info.getName().equals("Micrófono (Dispositivo de High "))
        {
                mixer = AudioSystem.getMixer(info);
        }
}
```

- Lectura de les dades d'entrada.
  S'estableix un buffer per realitzar les operacions de lectura de les dades d'entrada, i es crea un bucle en el que es llegirà el seu contingut per a fer el tractament posterior.
  Les dades s'emmagatzemen a una taula de tantes posicions com mostres (*samples*) s'estableixin, en aquest cas 44100. Per facilitar les proves s'ha decidit que aquests *samples* siguin de 8 bits, tot i que serà necessari fer-ho amb 16 per millorar la qualitat.

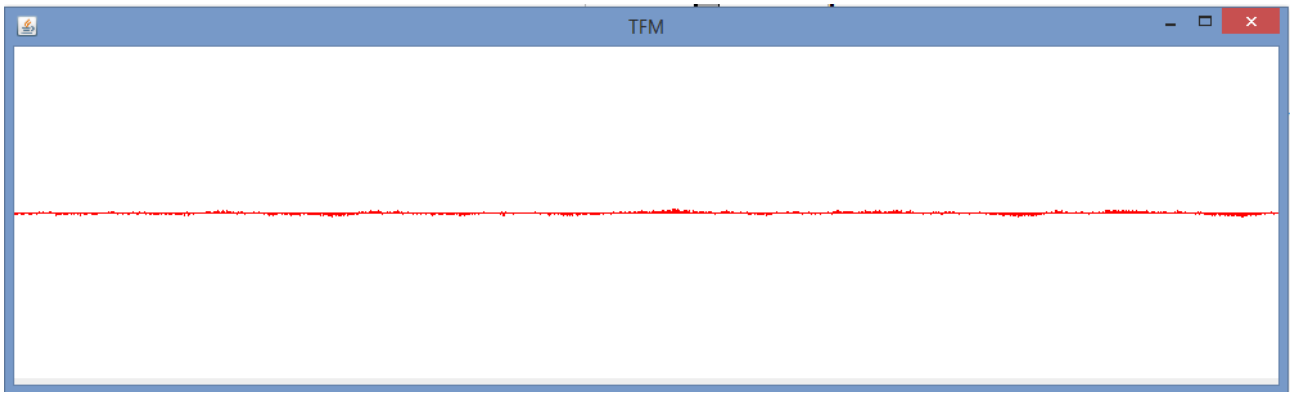
```
line = (TargetDataLine) AudioSystem.getLine(mixer.getTargetLineInfo()[0]);

line.open(audioFormat);

// Principal.
boolean stopped = false;
int numBytesRead;

line.start();
while (!stopped) {
        numBytesRead = line.read(data, 0, data.length);
        myApp.repaint();
}
```

- Tractament.
  S'ha realitzat un tractament simple de les dades per visualitzar-les gràficament i observar com varien.

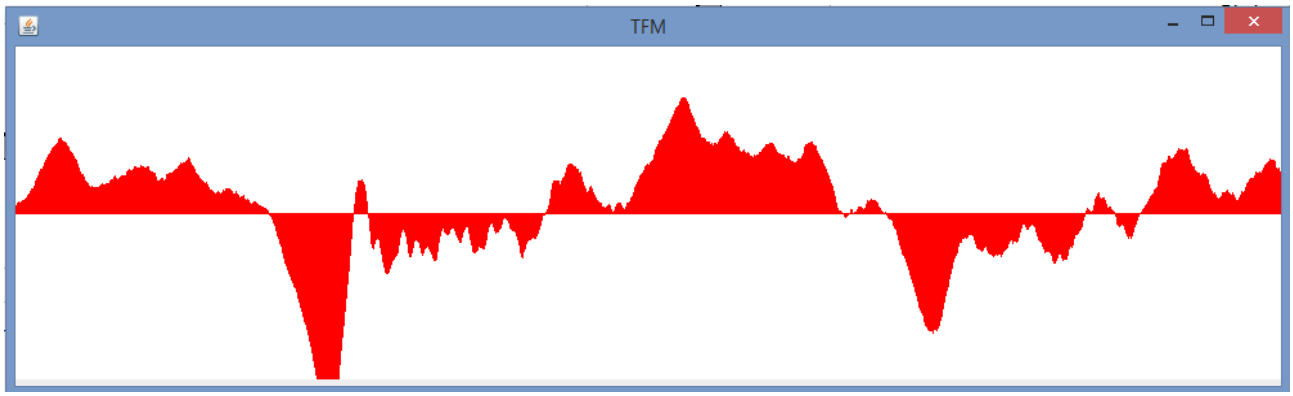
JOC DE PROVES

S'han realitzat diverses proves per constatar la variabilitat de les dades d'entrada del sistema.

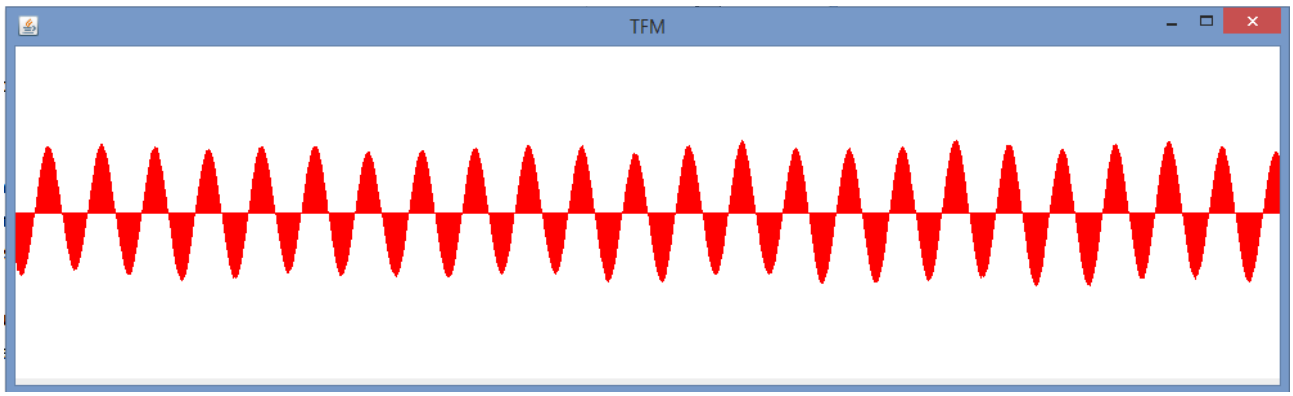
- So de fons.



- Pronunciació d'una vocal.



- Xiulada greu



- Xiulada aguda