

## La structure de l'application Digital-Scratch

Historique des versions	23/05/04 : Création du document
-------------------------	---------------------------------

### Sommaire

1 Introduction.....	2
2 Structure du système.....	3
2.1 Structure générale.....	3
2.1.1 Recorder.....	3
2.1.2 Analyser.....	3
2.1.3 Driver.....	4
2.1.4 Player.....	4
3 L'interface graphique.....	5

# 1 Introduction

Digital-Scratch est une application destinée à piloter un lecteur de fichier audio (ici AlsaPlayer; <http://www.alsaplayer.org>) via une platine vinyle. Tout les mouvements produits par l'utilisateur sur le disque seront analysés et retranscrit au niveau du player.

Il s'agit de développer tout le système allant de l'analyse du timecode du disque vinyle (via l'entrée ligne de la carte son) jusqu'au pilotage du player. Le système permettant l'analyse du timecode devra être totalement portable et non dépendant des drivers de la carte son et du player utilisé. On pourra parler en quelque sorte d'une librairie.

Les parties qui suivent expliquent comment sera structuré l'application. Chaque parties sera détaillées et un prototype de l'interface graphique sera élaboré.

## 2 Structure du système

### 2.1 Structure générale

Voici le schéma de la structure générale de l'application. Le système est découpé en 6 parties qui interagissent entre elles.

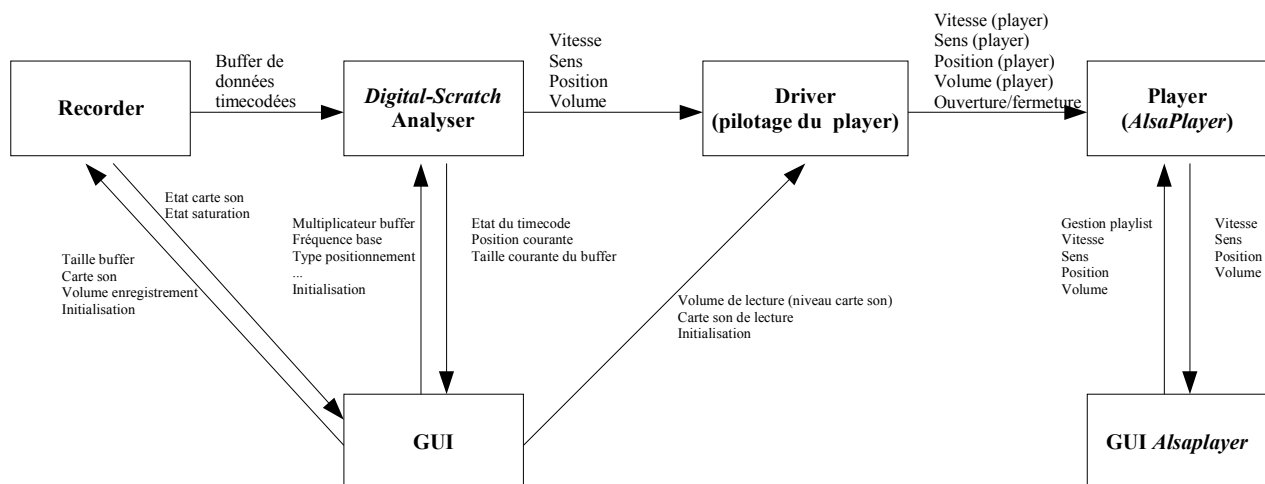


Illustration 1 - Structure générale de Digital-Scratch

#### 2.1.1 Recorder

Le **recorder** est l'entité qui va se charger de communiquer avec la carte son. Son rôle est de :

- initialiser la carte son
- modifier le volume d'enregistrement
- lire sur l'entrée ligne le flux audio (dans notre cas le timecode provenant du disque vinyle)

Il communique également avec l'interface graphique (**GUI**). Il l'a renseigne sur l'état courant de la carte son (initialisée ou non) et également sur la saturation du timecode. En effet, si le volume d'enregistrement du timecode est trop élevé, on risque d'enregistrer un son saturé, et peut être, perdre des informations importantes. C'est pourquoi, le module propose de piloter directement le volume de la carte son, via les fonctionnalités proposé par les drivers de carte son.

#### 2.1.2 Analyser

L'**analyser** est le coeur de *Digital-Scratch*. Il reçoit en entrée, les données représentant le timecode enregistré par le **recorder** et se charge d'en extraire les informations suivantes :

- la vitesse de rotation du disque
- le sens de rotation du disque
- la position de la cellule sur le disque
- le volume sonore estimé par rapport à la vitesse du disque

Il communique également avec la **GUI** en lui indiquant toutes les options de configurations du traitement du timecode :

- le multiplicateur de buffer de données
- la fréquence de base du timecode
- le type de positionnement souhaité (réel, relatif,...)
- ...

On pourra également réinitialiser l'**analyser** avec les nouvelles options depuis la **GUI**.

### 2.1.3 Driver

Le driver est l'élément intermédiaire entre l'**analyser** et le **player** permettant de traduire les notions de vitesse, sens, volume et position dans un langage compréhensible par le **player**.

Sa deuxième fonctionnalité est de gérer le lancement du **player** et également sa réinitialisation avec les paramètres de carte son appropriés. C'est pourquoi la **GUI** lui transmet la carte son utilisée pour la lecture et le volume sonore (celui de la carte et non celui du **player** qui ne sont pas forcément lié).

### 2.1.4 Player

Le dernier élément de la chaîne est le **player** équipé de sa **GUI** qui se charge de lire les fichiers audio présent dans sa playlist de la manière que lui impose le **driver**. Le pilotage du **player** depuis sa **GUI** sera dépendant du **player** choisi, ici *AlsaPlayer* ne propose que peu de fonctionnalités.

### 3 L'interface graphique

L'interface graphique du système se chargera de communiquer avec le **recorder**, l'**analyser** et le **driver**.

Pour chacune de ses entités on peut distinguer 2 types d'entrées. D'une part les entrées « statiques » qui correspondent aux entrées de l'utilisateur qui nécessitent un redémarrage du module voire du système total. Par exemple pour le cas du **recorder**, un changement de carte son nécessitera le redémarrage du système. D'autre part, il y a les entrées « dynamiques » qui peuvent se changer en temps réel, comme le volume d'enregistrement.

Il y a encore une dernière partie pour chaque entité qui se charge d'afficher l'état du système. Par exemple le **recorder** peut nous renseigner sur l'état courant de la carte son. Sur ces notions l'utilisateur n'a pas la main, ce n'est que de l'affichage.

Voici une petite maquette de l'interface graphique de *Digital-Scratch*.

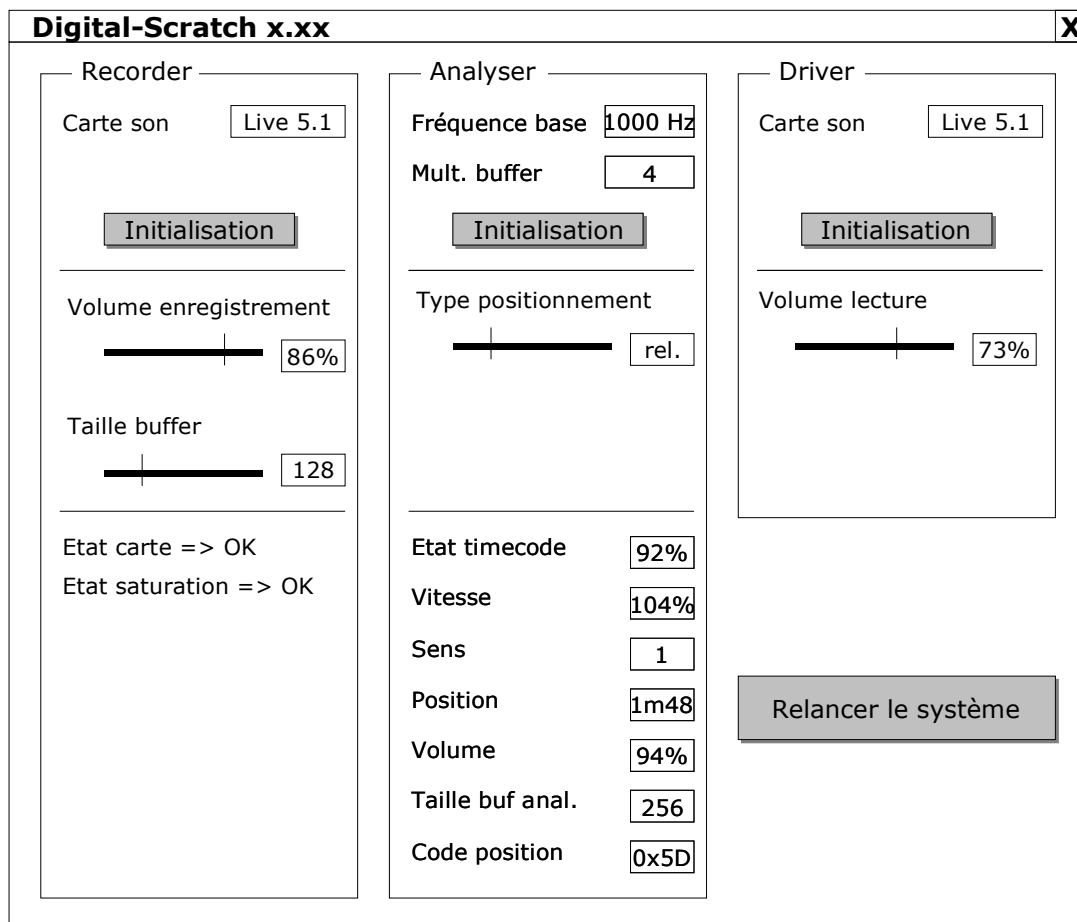
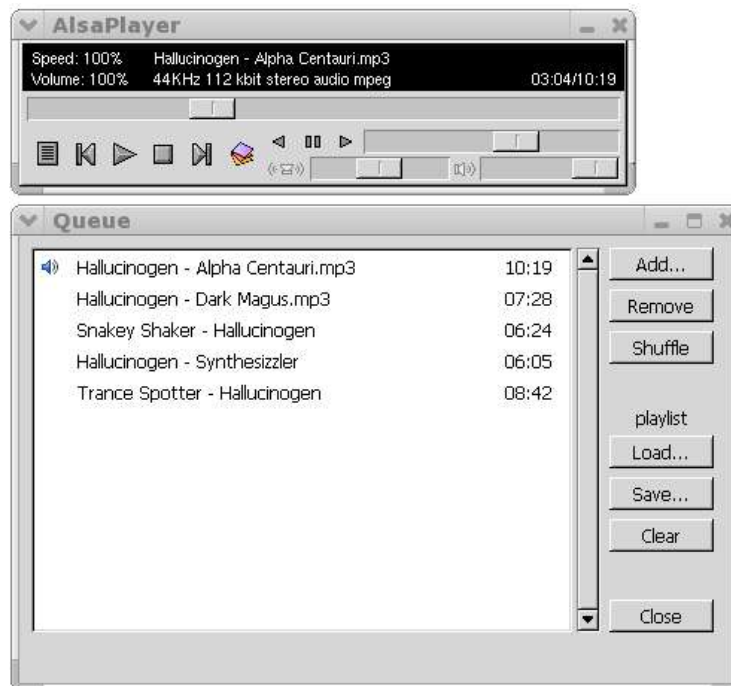


Illustration 2 - Maquette de l'interface graphique

L'interface graphique d'*AlsaPlayer* se présente sous la forme :



*Illustration 3 - Interface graphique d'AlsaPlayer*