

# Un exercice de programmation...

## Vintage...

### ...et un outil d'aide à la composition électroacoustique

Jean-Jacques Girardot & Jean-François Minjard

25 Mai 2022

# Le projet « Game Master »

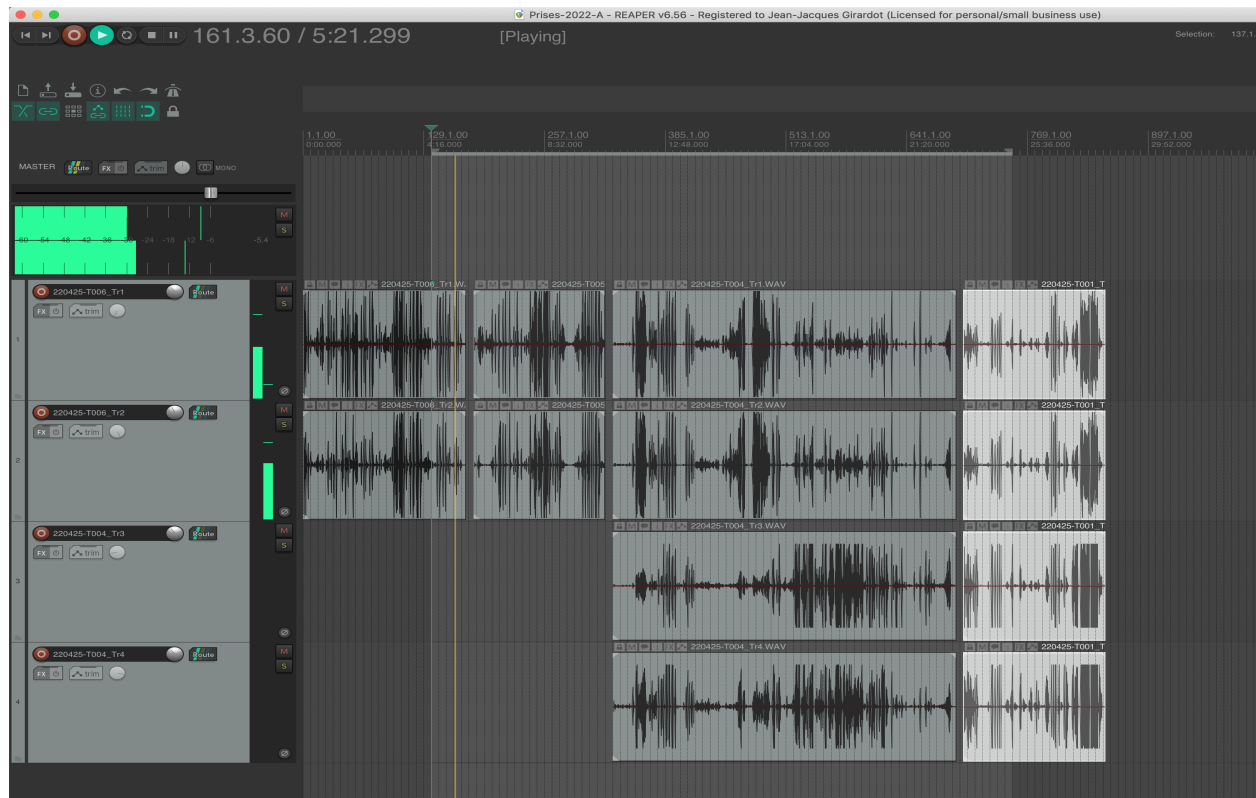
- Animer une installation électroacoustique
  - Jouer des « clips », petites séquences sonores
    - Utilisation d'un « DAW »
    - Choix de REAPER
      - Pistes son, avec les clips répartis sur la piste
  - Permettre l'interaction avec les « visiteurs »
    - Utilisation de capteurs
      - Piézos branchés sur des entrées de la carte son
    - Réactivité ?
      - Première version : un plugin (écrit en JS) lit la piste, conserve le son, le restitue lors de l'interaction

# Un DAW ?

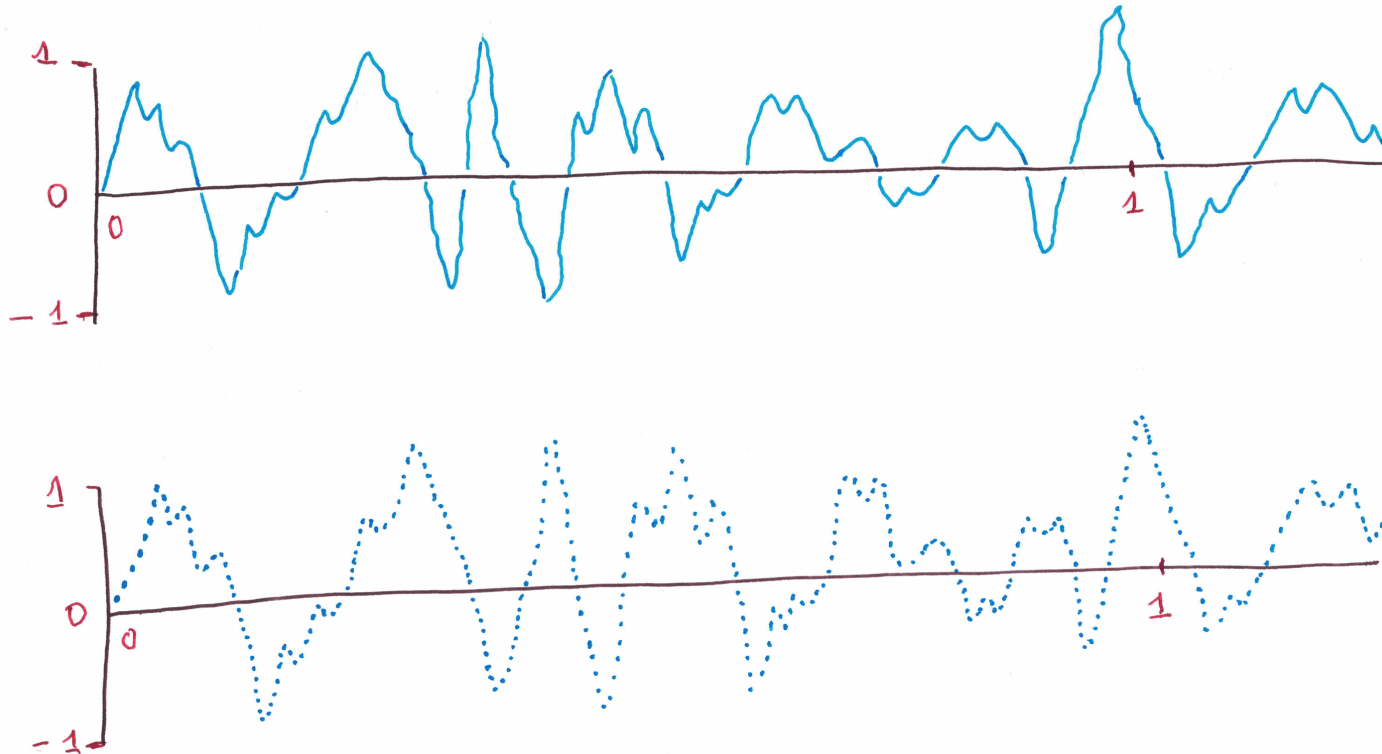
- Des « pistes » au sens d'un magnétophone sur lesquelles on place des fichiers audio
- Les pistes sont mixées vers une piste « master » qui peut être enregistrée ou dirigée vers une carte son
- Le « moteur » du DAW assure l'ensemble des traitements et des synchronisations
- Les pistes et fichiers peuvent recevoir des traitements audio relevant du « DSP », confiés à des « plug-ins » créés par des sociétés spécialisées
- Types de plug-ins standard : VST, AU
- Plug-ins propres à des DAW spécifiques : AAX, JSFX...

# L'outil

- REAPER : un «bon daw»
  - Puissant, économique, efficace, outil de création de scripts intégré, « vrai » multicanal



# Le son digital



- Une seconde : 44100/48000/96000 samples
- Si l'on a 25 pistes octophoniques, à 48kHz, avec un « calcul » par sample, on a droit à 0.1 microseconde par calcul...

# Poursuite du projet...

- Un premier besoin :
  - Créer des variations dans ces sons
    - Répétition des clips : uniformité, ennui...
    - Introduire du hasard dans le choix des clips
    - ... et dans la manière dont ils sont joués
      - Vitesse, volume, transposition
- Ecriture d'un script (sous forme de plugin) :
  - Lecture des clips depuis le disque
  - Variation de vitesse/hauteur à la manière d'un magnétophone

# Le langage de script de REAPER

- Ecriture de plugins traditionnels : C, C++, etc.
- Ou : utilisation de JS, propre à REAPER
  - Utilisé pour la première version de l'installation
  - Des avantages et des inconvénients...
- JS : « Jesus Sonic ».
  - Syntaxe de type « C simplifié »
  - Source, compilateur JIT, code machine non optimisé
  - Fournit « math.h », « string.h », quelques opérations de « DSP » et des fonctions graphiques

# Caractéristiques de JS

- Syntaxe de type C, les () remplacent les {}
- Constructions manquantes : switch/case, goto, for, macroprocesseur
- Type de données unique : le flottant double 64 bits
- Pas de déclaration de variables, identificateurs case-insensitive
- Fonctions non récurives, pas de référence avant possible
- Pas de bibliothèques autres que les fonctions de math.h, string.h, fft, convolution, quelques fonctions graphiques
- Accès à un tableau de 32M, propre à chaque plugin JSFX
- Accès à un tableau de 8M, partagé entre tous les plugins
- Accès à un nombre limité de chaînes de caractères
- Accès extérieurs filtrés : lecture dans certains répertoires, lecture et écriture contrôlée de préssets, pas d'écriture de fichiers.
- Excellente intégration à REAPER