Looper Mixer

Composer des sons à partir de samples

Joris Tillet – Bureau R220

26/03/2024

IN104 – ENSTA Paris

Sommaire

Présentation générale

Objectifs du cours

Gestion de projet en binôme :

- Respecter les dates imposées pour présenter un produit fonctionnel ;
- Choisir ses objectifs (imposés et sous votre initiative) ;
- Répartir le travail efficacement ;
- Apprendre et s'améliorer en programmation (en général, et en C).

Objectifs du projet

Minimum requis :

- Charger et jouer différents samples en boucle ;
- L'utilisateur peut choisir quels sons sont joués (simultanément) ;
- Les sons joués ensemble sont automatiquement synchronisés en temps.

Fonctionnalités supplémentaires

IHM (Interface Homme-Machine) :

- Afficher les samples disponibles et ceux en cours de lecture ;
- L'utilisateur peut activer ou désactiver des sons en cliquant dessus avec la souris et/ou en appuyant sur une touche du clavier.

Autres possibilités (non exhaustif) :

- Ajouter un nouveau type de samples qui ne sont joués qu'une seule fois (sans bouclage) ;
- Créer des samples depuis l'entrée micro ;
- Modifier le bpm global ;
- Modifier le volume par sample ;
- Modifier le pitch par sample ;
- Sauvegarder le mix créé ;
- Ajouter des effets (fade-in, fade-out, ...) ;
- Afficher le spectre ou une animation en fonction du son joué.

Gestionnaire de version (Git)

Intérêt de Git
But : suivre la vie d'un ou plusieurs fichiers.

- Quand le fichier a été modifié ?
- Quels sont les changements ?
- Pourquoi ces changements ?
- Qui en est à l'origine ?

$\Rightarrow$ Permet d'avoir plusieurs versions d'un même fichier, mais bien organisées.

Définition (Wikipédia)
Git : Git est un logiciel (libre) de gestion de versions décentralisé.

Définition (Wikipédia)
GitHub : GitHub est un site d'hébergement et de gestion de développement compatible avec Git.

- Git est un "VCS" (Version Control System).
- GitHub est le site de dépot le plus répandu, mais il en existe d'autre (GitLab, FramaGit, etc).

Installation de Git

Sur Linux (ou MacOS)
Git est disponible sur les dépots officiels :

```
$ sudo apt install git # Pour Ubuntu et dérivées
```

Sinon voir sur https://git-scm.com/downloads.

Sur Windows
Un installeur est disponible sur le site https://gitforwindows.org/.

- Il existe des outils graphiques (Git Kraken, QGit, GitHub, etc).
- Certains éditeurs de texte ou IDE intègrent directement les VCS.

Création d'un répertoire Git
Soit à partir d'un dossier déjà existant (en local) :

```
$ git init
```

Soit à partir d'un projet existant déjà :

```
$ git clone <adresse_du_projet>
```

Status

```
$ git status
```

Indique l'état actuel du répertoire :

- Les nouveaux fichiers pas encore suivis,
- les changements non enregistrés,
- d'autres informations utiles sur ce que vous êtes en train de faire.

Add

Ajout d'un fichier

```
$ git add <chemin>
```

ou

```
$ git add -A
```

Ajoute les fichiers de chemin à l'index de Git. L'option '`-A`' (ou `--all`) permet d'ajouter tous les fichiers du répertoire courant.

Commit

```
$ git commit -m "message du commit"
```

- Enregistre les changements avec une explication associée,
- L'option '-m' permet d'écrire le message obligatoire associé,
- L'option '-a' permet d'ajouter les modifications des fichiers déjà dans l'index avant le commit (évite le `git add`).

Log : montre les derniers commits

```
$ git log # options utiles : --oneline --color --graph
$ git whatchanged
$ git show
```

Création d'un *remote*

```
$ git remote add <nom_remote> <url>
```

Si un `git clone` a été utilisé au début pour initialiser le répertoire git, alors la *remote* existe déjà sous le nom *origin*.

Push

```
$ git push <nom_remote> <nom_branche>
```

- Met à jour le dépot distant.
- Il faut bien sûr avoir les droits sur ce dépot, et cette commande demande souvent une identification (compte GitHub).

Pull

```
$ git pull <nom_remote> <nom_branche>
```

- Met à jour le répertoire avec le dépot distant associé à *nom_remote*.
- Exécute en fait les deux commandes `git fetch` qui télécharge les données et `git merge` qui les fusionne avec le répertoire courant.
- Peut entraîner des conflits qu'il faut résoudre soi-même.

Paramètrage de Git

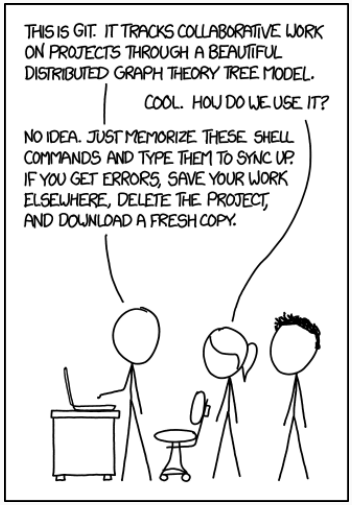
```
$ git config --global user.name "<name>"
$ git config --global user.email "<email@domain.fr>"
$ git config --global http.proxy <ip.address:port>
```

Un bon IDE

- **Obligatoire**,
- Exemples : Sublime, Atom, VS Code, JetBrains suite, etc...

Conclusion Git

- Très bonne gestion de versions,
- Compliqué au début,
- Meilleur ami ensuite.



Source : xkcd.com

Librairie SDL

SDL (*Simple DirectMedia Layer*)

- Librairie écrite en C ;
- Surcouche pour utiliser les différentes ressources média de votre machine ;
- Permet d'ouvrir des fenêtres et afficher des images et figures, de récupérer les évènements liés à l'appuie sur une touche du clavier ou de la souris, de jouer des sons, etc.

Installation SDL

Plusieurs possibilités :

- Directement sur Linux :
  ```
  $ sudo apt install libsdl2-dev libsdl2-mixer-dev
  ```
- Sur WSL : similaire à Linux,
- Sur une machine virtuelle : déjà installée.

Conclusion

Objectifs du jour

À l'issu de la séance d'aujourd'hui, vous devrez avoir votre environnement de travail prêt pour pouvoir démarrer le projet ensuite.

- Un compte GitHub par personne,
- Un *repository* créé par groupe pour le projet avec le ReadMe (les autres membres du groupe ajoutés en tant que collaborateur),
- Git installé dans votre environnement, configuré et synchronisé avec GitHub,
- SDL2 installé dans votre environnement,
- Tout le monde peut compiler et exécuter un exemple simple utilisant la SDL.

Questions ?