Pure Data & GEM

Pure Data est un environnement de programmation graphique qui permet de créer des applications multimédias en connectant des blocs de construction visuels appelés **objets**. Ces objets peuvent être utilisés pour manipuler des signaux audio, vidéo, et de données en temps réel.

Fonctionnalités

- **Audio Temps Réel :** Pure Data permet le traitement audio en temps réel avec une faible latence, adapté aux performances live.
- **Synthèse Sonore :** Création de synthétiseurs personnalisés, effets audio, processeurs de signal.
- **Contrôle MIDI :** traitement des données MIDI pour interagir avec des instruments électroniques et des contrôleurs MIDI.
- **Traitement Vidéo :** avec GEM et d'autres extensions, manipulation de vidéos et création d'installations visuelles.
- **Interaction et Interfaces :** intégration de capteurs, interfaces physiques, et réseaux pour des installations interactives visuelles et sonores

GEM

Voir GEM

Concepts de Base

1. **Patches :**

    1. Dans Pure Data, vous créez des **patches**, qui sont des fichiers visuels représentant votre programme.
    2. Un patch est constitué d'une collection d'objets connectés par des **câbles** pour former un réseau de traitement de données.

2. **Objets :**

    1. Chaque élément de traitement dans Pure Data est un **objet**. Il existe différents types d'objets pour des tâches variées, comme le traitement audio, la gestion des messages MIDI, la manipulation de données, etc.
    2. Vous pouvez créer un objet en tapant son nom dans un patch. Par exemple, `[osc~]` pour un oscillateur.

3. **Messages :**

    1. Les objets échangent des **messages** pour communiquer. Ces messages peuvent contenir des données numériques, des chaînes de caractères, ou des commandes.
    2. Les messages sont souvent déclenchés par des événements interactifs comme des clics ou des entrées utilisateur.

4. **Signal Audio :**

    1. Les signaux audio sont représentés par des câbles **tilde** (`~`), qui indiquent le flux de traitement audio en temps réel.
    2. Par exemple, `[osc~ 440]` génère un signal audio sinusoïdal à 440 Hz.

5. **Graphisme / vidéo:**

    1. Pure Data prend en charge le rendu graphique grâce à l'extension **GEM** (Graphics Environment for Multimedia), permettant la création d'animations et d'installations visuelles interactives.

6. **Bibliothèques complémentaires et extensions**

    1. Les fonctions de Pure Data peuvent être étendues de différentes manières, des patches réalisant certaines fonctions (*abstractions*), de nouveaux objets (*externals*) ou des bibliothèques de fonctions (*libraries*). Ces extensions peuvent être installées grâce à Deken accessible depuis le menu aide / installer des objets complémentaires

Variantes

- **PD Vanilla** : version de base, qui peut s'accompagner de bibliothèques d'objets additionnelles (type GEM) ou d'abstractions (assemblages d'objets pure data pour réaliser une fonction particulière) comme rjlib
- **PD-Extended :** Une version enrichie avec de nombreuses bibliothèques supplémentaires, mais cette version n'est plus mise à jour.
- **Purr Data (Pd-L2Ork) :** Une version moderne avec une interface graphique améliorée
- **Plug data** qui permet de compiler des patchs pure data pour en faire des plugins pour logiciels de musique
- **libpd** : intégration dans différents langages de prog , entre autre pour android et iOS
- **webpd** (pour réaliser des instruments dans le navigateur)
- **mobmuplat** (réaliser simplement des apps pour android et iOS)

Extensions

- **automatonism** : synthétiseur modulaire réalisé en pure data
- **rjlib** : une collection d'abstractions très pratiques pour le son

Liens

- pure data «vanilla» https://puredata.info/
- pd-extended https://puredata.info/
- purr data https://agraef.github.io/purr-data/
- plugdata https://plugdata.org/
- libpd : https://github.com/libpd/libpd
- webpd : https://github.com/sebpiq/WebPd
- mobmuplat : https://danieliglesia.com/mobmuplat/
- automatonism : https://www.automatonism.com/
- rjlib : https://github.com/rjdj/rjlib

Documentation

Documentation intégrée dans chaque objet : clic droit / aide, ou dans le menu aide : navigateur d'aide, liste des objets, etc. Le navigateur d'aide affiche aussi les fichiers d'aide associés aux abstractions installées par deken, manuellement ou par apt (sur linux)

- http://msp.ucsd.edu/Pd_documentation/
- https://pd.iem.sh/objects/

Apprendre pure data

- http://www.pd-tutorial.com/

Livres

“Flossmanual pure data” par collectif **“Designing Sound”** par Andy Farnell - Un livre sur le design sonore avec Pure Data. “Multimedia programming with pure data”

Liens

- https://forum.pdpatchrepo.info
- https://github.com/MikeMorenoDSP/awesome-puredata
- https://patchstorage.com/
- http://codelab.fr/73

- composition générative : https://github.com/emoc/pure-data_tactique
- interaction avec arduino et processing : https://github.com/emoc/arduino_interaction

Réaliser des externals

- https://agraef.github.io/pd-lua/
- https://github.com/pure-data/externals-howto

Article extrait de : http://www.lesporteslogiques.net/wiki/ - **WIKI Les Portes Logiques**
Adresse : **http://www.lesporteslogiques.net/wiki/ressource/logiciel/pure-data/start**
Article mis à jour: **2024/07/28 23:12**