



## Intégration des Logs dans le Projet

### Système de journalisation (`log.c` / `log.h`)

#### Objectif

Le système de journalisation (logging) a été intégré pour :

- Tracer les événements clés du programme.
- Faciliter le débogage.
- Comprendre les étapes de traitement du réseau de neurones et de l'interface SDL.

#### `log.h` – Interface des fonctions de log

Ce fichier contient :

- L'énumération des niveaux de log (`LOG\_LEVEL\_INFO`, `LOG\_LEVEL\_WARNING`, `LOG\_LEVEL\_ERROR`).
- La fonction générique `log\_message(level, format, ...)` pour afficher des logs formatés.
- Des fonctions utilitaires pour centraliser les messages liés à certaines actions :
  - `log\_colorier\_once(const char \*mode)` – loggue une seule fois le coloriage (R/B ou RGB).
  - `log\_generation\_spirale(const char \*type)` – loggue la génération de spirales.
  - `log\_save\_success()` / `log\_save\_failure()` – logs liés à la sauvegarde.
  - `log\_load\_success()` / `log\_load\_failure()` – logs liés au chargement.
  - `log\_choix\_train\_load()` – loggue l'ouverture de la fenêtre "Train/Load".

#### `log.c` – Implémentation centralisée

Ce fichier implémente :

1. `log\_message()` : utilise `va\_list` pour formater un message avec horodatage et niveau.
2. Fonctions `log\_\*\_once()` : utilisent `static int deja\_fait = 0` pour n'afficher le message qu'une seule fois.
3. Tous les logs sont centralisés pour une traçabilité uniforme.

#### Intégration dans le projet

Contexte	Fichier	Log appelé
Propagation dans le réseau	neural_network.c	log_propagation_once()
Backpropagation	neural_network.c	log_retropropagation()
Colorier écran (R/B ou RGB)	data.c	log_colorier_once("R/B") / ("RGB")



<b>Génération de spirales</b>	data.c	log_generation_spirale("R/B") / ("multi-classes")
<b>Sauvegarde du réseau</b>	save.c	log_save_success() / log_save_failure()
<b>Chargement d'un fichier</b>	save.c	log_load_success() / log_load_failure()
<b>Affichage Train ou Load</b>	ui_manager.c	log_choix_train_load()

☒ **Résultat**

- Les logs permettent de comprendre l'état du programme sans ouvrir GDB.
- Le système est minimaliste, centralisé et facilement réutilisable.
- Il améliore significativement la traçabilité des actions utilisateur et du réseau.