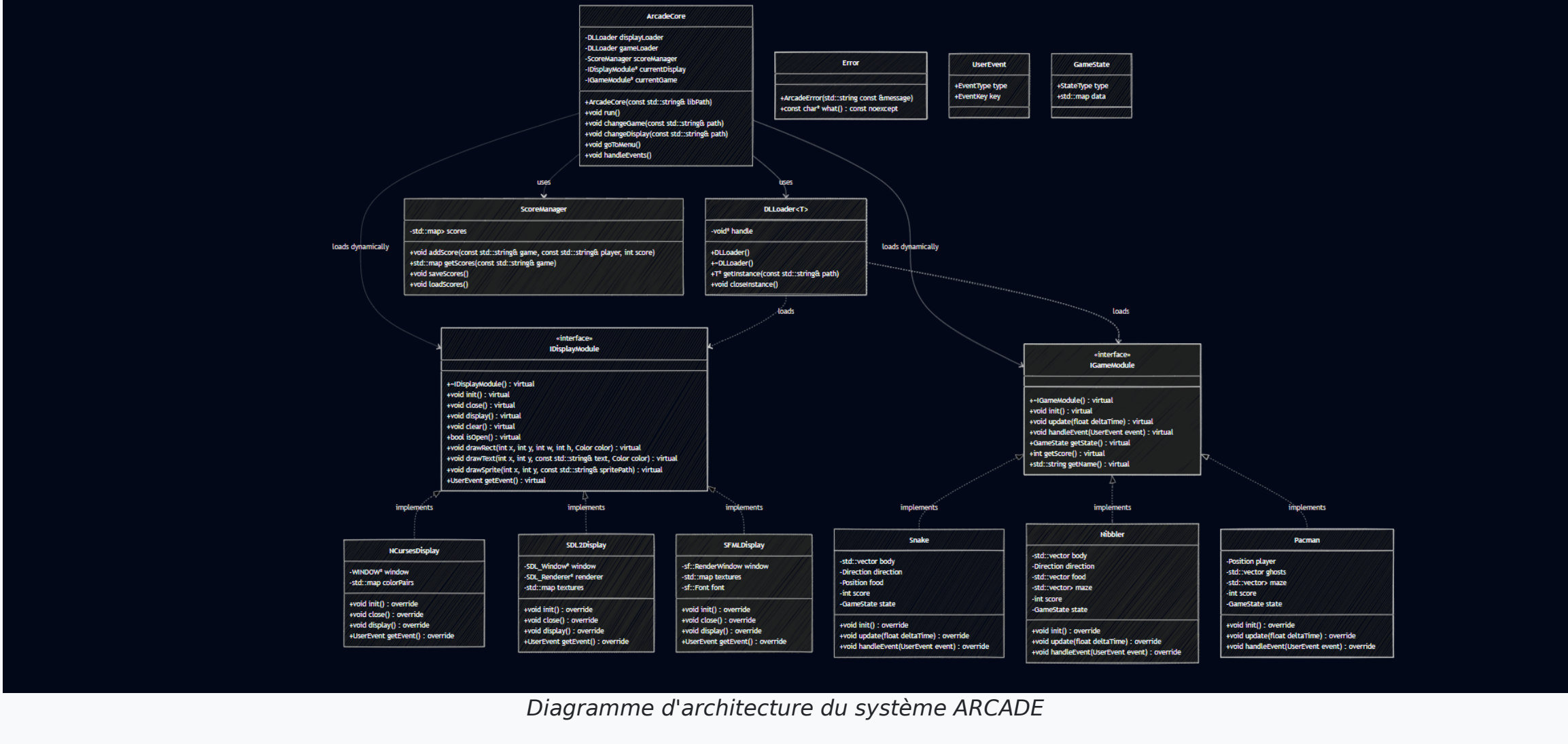


Vue d'ensemble

Le cœur du système ARCADE est conçu autour d'un principe de modularité et de chargement dynamique. Cette architecture permet de charger et décharger des bibliothèques graphiques et des jeux à l'exécution, sans avoir à recompiler le programme principal.



Composants principaux

⚙️ ArcadeCore

Le composant central qui gère tout le système. Il est responsable de:

- Détecter et charger les bibliothèques disponibles
- Gérer les états du programme (menu, jeu, game over)
- Intercepter et acheminer les entrées utilisateur
- Basculer entre les différentes bibliothèques graphiques et jeux
- Maintenir les scores

```
class ArcadeCore {
public:
    ArcadeCore(const std::string &graphicLibPath);
    ~ArcadeCore();

    void run();

private:
    // États du core
    enum class CoreState {
        MENU,
        GAME,
        GAME_OVER,
        EXIT
    };

    // Méthodes principales
    void loadLibraries();
    void displayMenu();
    void handleMenuInputs();
    void updateGame();
    void renderGame();

    // Autres membres et méthodes...
};
```

[Copy](#)

🔧 DLLoader

Le chargeur de bibliothèques dynamiques qui encapsule les fonctions dlopen, dlsym, dlclose pour:

- Charger les bibliothèques dynamiques au runtime
- Récupérer les symboles depuis les bibliothèques
- Instancier les modules (graphiques ou jeux)
- Décharger proprement les bibliothèques

```
template
class DLLoaderImpl {
public:
    DLLoaderImpl();
    ~DLLoaderImpl();

    bool load(const std::string &path);
    void unload();
    T *getInstance();

private:
    void *_handle;
    T *_instance;
    // Autres membres...
};
```

[Copy](#)

📅 EventManager

Gestionnaire d'événements qui:

- Capture les entrées utilisateur (clavier, souris)
- Les traduit en événements génériques pour les modules
- Gère l'état des touches (pressée/relâchée)

🏆 ScoreManager

Gestionnaire de scores qui:

- Enregistre et charge les meilleurs scores
- Trie et affiche les scores par jeu
- Persiste les données entre les sessions

-  [Accueil](#)
-  [Démarrage](#)
-  [Architecture](#)
-  [Biblio. Graphiques](#)
-  [Créer un jeu](#)
-  [API](#)
- ☐

[Mode sombre](#)