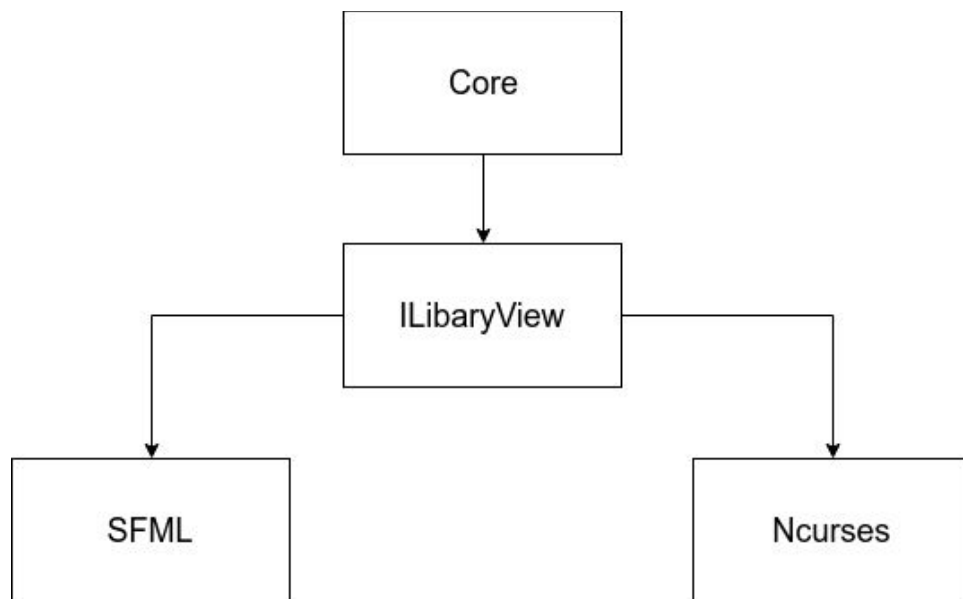


DOCUMENTATION

Dans cette documentation nous allons expliquer comment ajouter une nouvelle librairie graphique compatible avec notre système.

Notre programme possède un Core, qui va nous permettre de lancer le menu et le jeu avec la librairie graphique de votre choix en exécutant le programme. En changeant de librairie pendant que le programme est en cours d'exécution, il va automatiquement charger celle-ci.

En effet, nous utilisons une interface appelée ILibraryView pour que chaque librairie graphique utilise les mêmes fonctions, ce qui simplifie donc l'appel des méthodes depuis le Core.



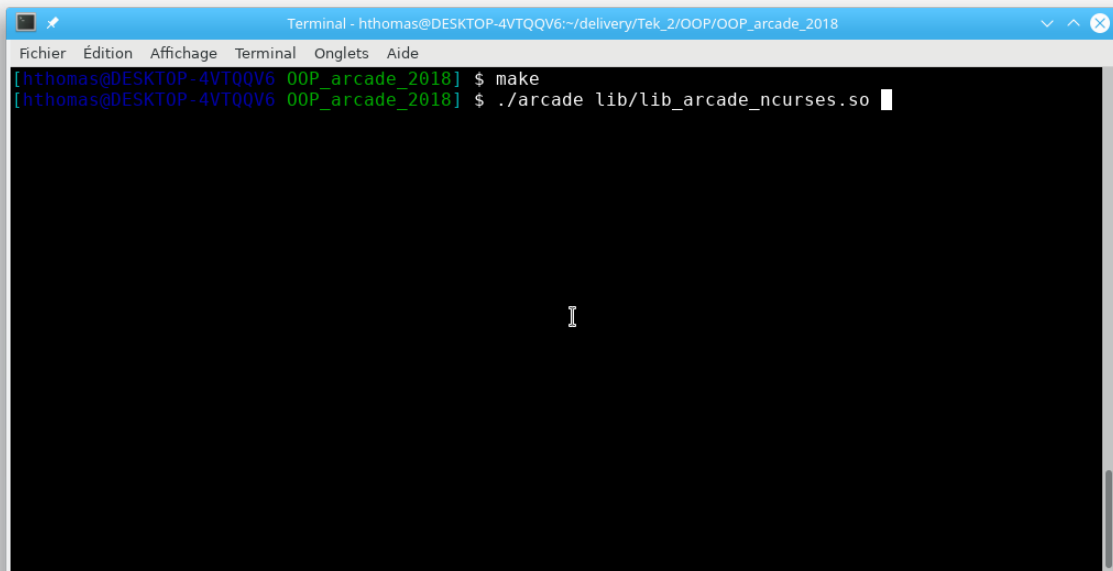
Suivant la librairie utilisée, le Core appellera toujours la méthode de l'interface.

Par exemple, la méthode qui se nomme "putText", permet d'afficher le texte souhaité dans le menu ou bien le jeu. En appelant cette fonction dans le Core, le texte s'affichera en SFML et Ncurses.

Ce qui veut dire que vous pourrez ajouter une nouvelle librairie, à condition d'utiliser les mêmes méthodes de l'interface.

Pour pouvoir ajouter votre librairie il suffit de le compiler en librairie dynamique (qui est un fichier de type “so”) dans le dossier “lib” se trouvant dans la racine du projet. Ainsi, le Core se chargera de la récupérer et lancera votre librairie automatiquement.

Exemple d'utilisation :



```
Terminal - hthomas@DESKTOP-4VTQQV6:~/delivery/Tek_2/OOP/OOP_arcade_2018
Fichier  Édition  Affichage  Terminal  Onglets  Aide
[hthomas@DESKTOP-4VTQQV6: OOP_arcade_2018] $ make
[hthomas@DESKTOP-4VTQQV6: OOP_arcade_2018] $ ./arcade lib/lib_arcade_ncurses.so
```

Liste des méthodes de l'interface :

```
class ILibraryView {
public:
    virtual ~ILibraryView() = default;

    virtual void init(Arcade::ICore *core) = 0;
    virtual void close() = 0;

    virtual void clear() = 0;
    virtual void refresh() = 0;

    virtual std::string getCharKeyboardEvent() = 0;
    virtual Arcade::Command getCommand() = 0;
    virtual void putText(const std::string text, Color color,
        std::size_t x, std::size_t y, std::size_t size) = 0;
    virtual void assignTexture(const std::string path, cellTileType type, Color color) = 0;
};
```