How does the game works ? Inside the main menu you can select the graphical library you want to use by using the **arrow** and **enter** when the one you want is hovered. Switch over the games using **tab** and over the graphical libraries using **space bar**

To create a new graphical library:

- Create a Api.cpp that will be used as an entry point, you will need to define :
    - Arcade::Graph::IDisplayModule *getDisplayModule();** -> Create an instance of your display module that inherit from IDisplayModule and return it
    - void destroyDisplayModule(Arcade::Graph::IDisplayModule *displayModule);** -> Free all ressources linked to this displayModule
    - const char *getName();** -> Return the name of the library
    - LibType getType(); -> Return the type of the library, here GRAPH
- Then implement the IDisplayModule interface in your library, your class will act as the manager:
    - The update fonction is called each frames, this is where you add how you want to display the entities
        - !!! Pay attention to IMusic, ISprite and IText that are the three type of components that contains data used by the graphical libraries

To create a new game library:

- Create a Api.cpp that will be used as an entry point, you will need to define :
    - Arcade::Game::IGameModule *getGameModule();** -> Create an instance of your game module that inherit from IGameModule and return it
    - void destroyGameModule(Arcade::Game::IGameModule *gameModule);** -> Free all ressources linked to this IGameModule
    - const char *getName();** -> Return the name of the library
    - LibType getType(); -> Return the type of the library, here GAME
- Then implement the IGameModule interface in your library, your class will act as the manager:
    - The IGameModule is your SceneManager, you have to create Scene that contains an EntityManager and a SystemManager, pay attention to the documentation provided for these parts
    - The *update* fonction is called each frames, this is where you add how you want to display the entities
        - !!! Pay attention to IMusic, ISprite and IText that are the three type of components that contains data used by the graphical libraries
    - Arcade::ECS::IEntityManager &getCurrentEntityManager() -> Returns the currently used entityManager inside the currently used Scene