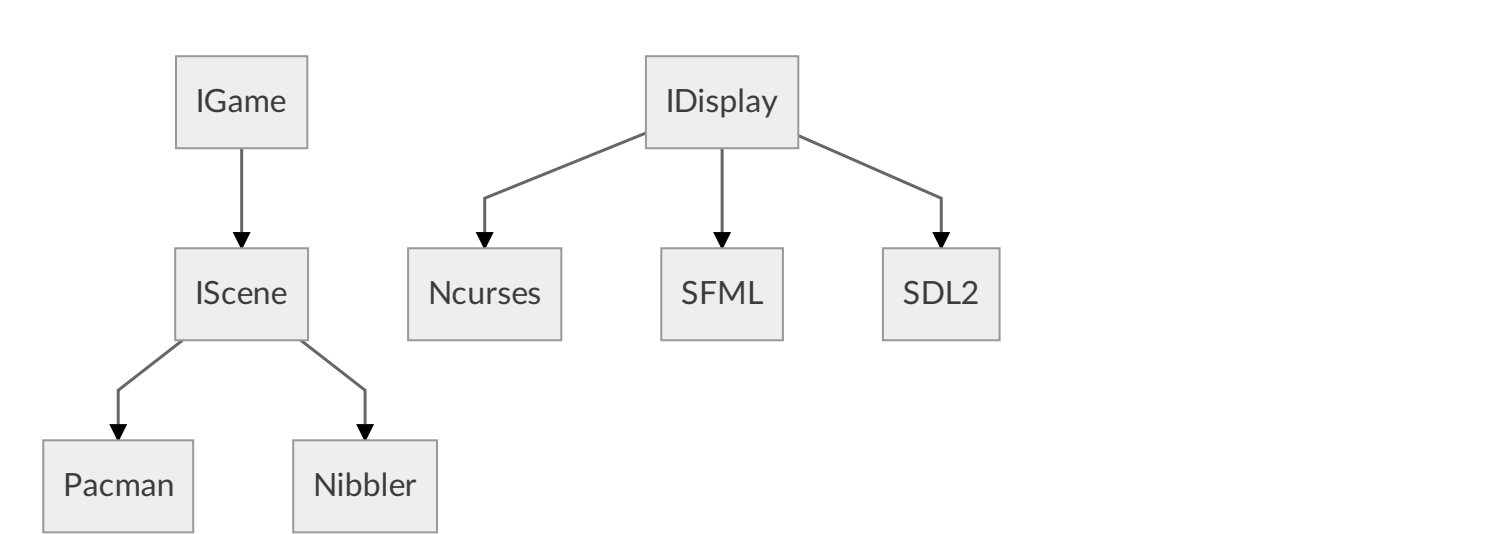


Documentation for Arcade Project

Welcome to Arcade Project

Arcade is a gaming platform: a program that lets the user choose a game to play and the graphic library to use.

Class Diagram



Useful function and macros

arcade::TypeEvent is an enumeration that contains the different types of events that the library can have, for example arcade::WINDOW, which corresponds to an event in the window.

Here is the TypeEvent declaration that is in the arcade namespace:

```
namespace arcade {
    enum TypeEvent {
        WINDOW,
        JOYSTICK,
        KEYBOARD,
        MOUSE
    };
}
```

Texture is a structure that will allow the exchange of information concerning the textures to be displayed as well as the texts and the characters.

Here is the Texture & Position declaration:

```
struct Position {
    int x;
    int y;
};

struct Texture {
    std::string path;
    char similar;
    bool isFile;
    bool display;
    Position position;
};
```

NOTE:

- The **path** variable is the path of the image, **it must be** either **relative** to the arcade executable or **complete**.
- The **similar** variable is used to **replace the texture with a char** if the current graphic library doesn't support images.
- The **isFile** variable is set to true if is a file, otherwise **path** is use as a text.
- If the **display** variable is set to true then the texture is displayed, otherwise the texture is not loaded.
- The **position** variable is the position of the **top-left** corner of the **Texture**.

Add your Graphics Library

Tips: Your graphics library must be in the folder **lib/** on the root of the project.

Your class must inherit from the IDisplay class. Here is the abstract class IDisplay :

```
class IDisplay {
public:
    virtual ~IDisplay() = default;
public:
    virtual bool Display() = 0;
    virtual bool isKey() = 0;
    virtual bool isOpen() = 0;
    virtual bool GetKey(arcade::TypeEvent typeEvent, std::string const &eventName) = 0;
    virtual bool loadTexture(std::map<std::string, Texture> const &textureList) = 0;
    virtual bool loadMap(std::vector<std::vector<char>> const &mapCharacter) = 0;
    virtual bool loadText(std::map<std::string, Texture> const &textureList) = 0;
    virtual void destroyWindow() = 0;
    virtual void changeLibrary(std::string const &path) = 0;
    virtual bool getChange() const = 0;
    virtual std::string const &getNewGamePath() const = 0;
    virtual void setNewGamePath(std::string const &path) = 0;
    virtual bool getSwitchScene() const = 0;
    virtual void setSwitchScene(bool state) = 0;
    virtual void setChange(bool state) = 0;
    virtual std::string const &getLibraryPath() const = 0;
protected:
    bool change;
    bool switchScene;
    std::string newGamePath;
    std::string newLibraryPath;
};
```

Note:

- If an exception needs to be thrown, use the **arcade::GraphicsLibraryError** class.
- All the events that can be used are available in the utils.hpp file.

Add your Game

Tips: Your game library must be in the folder **games/** on the root of the project.