

Arcade

Generated by Doxygen 1.9.7

1 Namespace Index	1
1.1 Namespace List	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	5
3.1 Class List	5
4 File Index	7
4.1 File List	7
5 Namespace Documentation	9
5.1 Arcade Namespace Reference	9
5.1.1 Detailed Description	10
5.1.2 Enumeration Type Documentation	10
5.1.2.1 Keys	10
5.1.3 Variable Documentation	10
5.1.3.1 apple	10
5.1.3.2 game_nibbler	11
5.1.3.3 game_nibbler_selected	11
5.1.3.4 game_snake	11
5.1.3.5 game_snake_selected	11
5.1.3.6 game_title	11
5.1.3.7 grass	12
5.1.3.8 lib_ncurses	12
5.1.3.9 lib_ncurses_selected	12
5.1.3.10 lib_sdl	12
5.1.3.11 lib_sdl_selected	12
5.1.3.12 lib_sfml	13
5.1.3.13 lib_sfml_selected	13
5.1.3.14 lib_title	13
5.1.3.15 map_menu	13
5.1.3.16 snake_body_horizontal	13
5.1.3.17 snake_body_vertical	14
5.1.3.18 snake_head_east	14
5.1.3.19 snake_head_north	14
5.1.3.20 snake_head_south	14
5.1.3.21 snake_head_west	14
5.1.3.22 wall	15
6 Class Documentation	17
6.1 Arcade::Arc Class Reference	17
6.2 Assets Struct Reference	17

6.3 DLoader< T > Class Template Reference	18
6.4 Arcade::IGameLib Class Reference	18
6.4.1 Detailed Description	18
6.5 Arcade::IGraphicLib Class Reference	19
6.6 Arcade::MENU Class Reference	19
6.6.1 Member Function Documentation	20
6.6.1.1 change_libs()	20
6.6.1.2 collisions()	20
6.6.1.3 directions_assigner()	20
6.6.1.4 drawSnake()	20
6.6.1.5 end_game_score()	20
6.6.1.6 getName()	20
6.6.1.7 init()	21
6.6.1.8 map_creator()	21
6.7 Arcade::NCURSES Class Reference	21
6.7.1 Member Function Documentation	22
6.7.1.1 closeWindow()	22
6.7.1.2 drawText()	22
6.7.1.3 getLastEvent()	22
6.7.1.4 getName()	22
6.7.1.5 mapManager()	22
6.7.1.6 popWindow()	23
6.7.1.7 refresh()	23
6.7.1.8 select()	23
6.8 Arcade::NIBBLER Class Reference	23
6.8.1 Member Function Documentation	24
6.8.1.1 change_libs()	24
6.8.1.2 collisions()	24
6.8.1.3 directions_assigner()	24
6.8.1.4 drawSnake()	24
6.8.1.5 end_game_score()	24
6.8.1.6 getName()	25
6.8.1.7 init()	25
6.8.1.8 map_creator()	25
6.9 Arcade::SDL Class Reference	25
6.9.1 Member Function Documentation	26
6.9.1.1 closeWindow()	26
6.9.1.2 drawText()	26
6.9.1.3 getLastEvent()	26
6.9.1.4 getName()	26
6.9.1.5 mapManager()	26
6.9.1.6 popWindow()	27

6.9.1.7 refresh()	27
6.9.1.8 select()	27
6.10 Arcade::SFML Class Reference	27
6.10.1 Member Function Documentation	28
6.10.1.1 closeWindow()	28
6.10.1.2 drawText()	28
6.10.1.3 getLastEvent()	28
6.10.1.4 getName()	28
6.10.1.5 mapManager()	28
6.10.1.6 popWindow()	29
6.10.1.7 refresh()	29
6.10.1.8 select()	29
6.11 Arcade::SNAKE Class Reference	29
6.11.1 Member Function Documentation	30
6.11.1.1 change_libs()	30
6.11.1.2 collisions()	30
6.11.1.3 directions_assigner()	30
6.11.1.4 drawSnake()	30
6.11.1.5 end_game_score()	30
6.11.1.6 getName()	31
6.11.1.7 init()	31
6.11.1.8 map_creator()	31
7 File Documentation	33
7.1 Arcade.cpp	33
7.2 Arcade.hpp	34
7.3 Loader.hpp	34
7.4 IGameLib.hpp	35
7.5 Menu.hpp	36
7.6 Nibbler.hpp	36
7.7 Snake.hpp	37
7.8 IGraphicLib.hpp	38
7.9 NCurses.hpp	39
7.10 SDL.hpp	39
7.11 SFML.hpp	40
7.12 Assets.hpp	40
7.13 Keys.hpp	41
Index	43

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

Arcade	
Arcade project namespace	9

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Arcade::Arc	17
Assets	17
DLLoader< T >	18
Arcade::IGameLib	18
Arcade::MENU	19
Arcade::NIBBLER	23
Arcade::SNAKE	29
Arcade::IGraphicLib	19
Arcade::NCURSES	21
Arcade::SDL	25
Arcade::SFML	27

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Arcade::Arc	17
Assets	17
DLLoader< T >	18
Arcade::IGameLib	
Game libraries virtual class	18
Arcade::IGraphicLib	19
Arcade::MENU	19
Arcade::NCURSES	21
Arcade::NIBBLER	23
Arcade::SDL	25
Arcade::SFML	27
Arcade::SNAKE	29

Chapter 4

File Index

4.1 File List

Here is a list of all documented files with brief descriptions:

src/Core/ Arcade.cpp	33
src/Core/ Arcade.hpp	34
src/Core/Loader/ Loader.hpp	34
src/Games/ IGameLib.hpp	35
src/Games/Menu/ Menu.hpp	36
src/Games/Nibbler/ Nibbler.hpp	36
src/Games/Snake/ Snake.hpp	37
src/Graphics/ IGraphicLib.hpp	38
src/Graphics/NCURSES/ NCurses.hpp	39
src/Graphics/SDL/ SDL.hpp	39
src/Graphics/SFML/ SFML.hpp	40
src/include/ Assets.hpp	40
src/include/ Keys.hpp	41

Chapter 5

Namespace Documentation

5.1 Arcade Namespace Reference

[Arcade](#) project namespace.

Classes

- class [Arc](#)
- class [IGameLib](#)
Game libraries virtual class.
- class [IGraphicLib](#)
- class [MENU](#)
- class [NCURSES](#)
- class [NIBBLER](#)
- class [SDL](#)
- class [SFML](#)
- class [SNAKE](#)

Enumerations

- enum [Keys](#) {
 NONE , **A** , **B** , **C** ,
 D , **E** , **F** , **G** ,
 H , **I** , **J** , **K** ,
 L , **M** , **N** , **O** ,
 P , **Q** , **R** , **S** ,
 T , **U** , **V** , **W** ,
 X , **Y** , **Z** , **LEFT** ,
 RIGHT , **UP** , **DOWN** , **ENTER** ,
 SPACE , **DELETE** , **BACKSPACE** , **TAB** ,
 ESC , **MOUSELEFT** , **MOUSERIGHT** }

Functions

- [Arcade::IGameLib](#) * **getInstance** ()
- int **chooseNumber** (int min, int max)

Variables

- [Assets map_menu](#)
- [Assets lib_title](#)
- [Assets game_title](#)
- [Assets game_snake](#)
- [Assets game_nibbler](#)
- [Assets lib_sdl](#)
- [Assets lib_ncurses](#)
- [Assets lib_sfml](#)
- [Assets lib_sdl_selected](#)
- [Assets lib_ncurses_selected](#)
- [Assets lib_sfml_selected](#)
- [Assets game_snake_selected](#)
- [Assets game_nibbler_selected](#)
- [Assets wall](#)
- [Assets grass](#)
- [Assets apple](#)
- [Assets snake_head_north](#)
- [Assets snake_head_south](#)
- [Assets snake_head_east](#)
- [Assets snake_head_west](#)
- [Assets snake_body_horizontal](#)
- [Assets snake_body_vertical](#)

5.1.1 Detailed Description

[Arcade](#) project namespace.

5.1.2 Enumeration Type Documentation

5.1.2.1 Keys

```
enum Arcade::Keys
```

All those keys should be handled by any graphic libraries or any games

5.1.3 Variable Documentation

5.1.3.1 apple

```
Assets Arcade::apple
```

Initial value:

```
{  
    .image_path = "assets/image/food_snake_sprite.png",  
    .character = "assets/text/food.txt",  
    .color = "green",  
    .width = 32,  
    .height = 32  
}
```


5.1.3.2 game_nibbler

Assets Arcade::game_nibbler

Initial value:

```
{
    .image_path = "assets/image/nibbler_menu_sprite.png",
    .character = "assets/text/game_nibbler.txt",
    .color = "red_black",
    .width = 128,
    .height = 32
}
```

5.1.3.3 game_nibbler_selected

Assets Arcade::game_nibbler_selected

Initial value:

```
{
    .image_path = "assets/image/nibbler_selected.png",
    .character = "assets/text/game_nibbler.txt",
    .color = "red",
    .width = 128,
    .height = 32
}
```

5.1.3.4 game_snake

Assets Arcade::game_snake

Initial value:

```
{
    .image_path = "assets/image/snake_menu_sprite.png",
    .character = "assets/text/game_snake.txt",
    .color = "red_black",
    .width = 128,
    .height = 32
}
```

5.1.3.5 game_snake_selected

Assets Arcade::game_snake_selected

Initial value:

```
{
    .image_path = "assets/image/snake_selected.png",
    .character = "assets/text/game_snake.txt",
    .color = "red",
    .width = 128,
    .height = 32
}
```

5.1.3.6 game_title

Assets Arcade::game_title

Initial value:

```
{
    .image_path = "assets/image/games_menu_sprite.png",
    .character = "assets/text/game_title.txt",
    .color = "yellow_black",
    .width = 192,
    .height = 64
}
```

5.1.3.7 grass

Assets Arcade::grass

Initial value:

```
{
    .image_path = "assets/image/grass_sprite.png",
    .character = "assets/text/grass.txt",
    .color = "green",
    .width = 32,
    .height = 32
}
```

5.1.3.8 lib_ncurses

Assets Arcade::lib_ncurses

Initial value:

```
{
    .image_path = "assets/image/ncurses_menu_sprite.png",
    .character = "assets/text/lib_ncurses.txt",
    .color = "red_black",
    .width = 128,
    .height = 32
}
```

5.1.3.9 lib_ncurses_selected

Assets Arcade::lib_ncurses_selected

Initial value:

```
{
    .image_path = "assets/image/ncurses_selected.png",
    .character = "assets/text/lib_ncurses.txt",
    .color = "red",
    .width = 128,
    .height = 32
}
```

5.1.3.10 lib_sdl

Assets Arcade::lib_sdl

Initial value:

```
{
    .image_path = "assets/image/sdl_menu_sprite.png",
    .character = "assets/text/lib_sdl.txt",
    .color = "red_black",
    .width = 128,
    .height = 32
}
```

5.1.3.11 lib_sdl_selected

Assets Arcade::lib_sdl_selected

Initial value:

```
{
    .image_path = "assets/image/sdl_selected.png",
    .character = "assets/text/lib_sdl.txt",
    .color = "red",
    .width = 128,
    .height = 32
}
```

5.1.3.12 lib_sfml

Assets Arcade::lib_sfml

Initial value:

```
{
    .image_path = "assets/image/sfml_menu_sprite.png",
    .character = "assets/text/lib_sfml.txt",
    .color = "red_black",
    .width = 128,
    .height = 32
}
```

5.1.3.13 lib_sfml_selected

Assets Arcade::lib_sfml_selected

Initial value:

```
{
    .image_path = "assets/image/sfml_selected.png",
    .character = "assets/text/lib_sfml.txt",
    .color = "red",
    .width = 128,
    .height = 32
}
```

5.1.3.14 lib_title

Assets Arcade::lib_title

Initial value:

```
{
    .image_path = "assets/image/libraries_menu_sprite.png",
    .character = "assets/text/lib_title.txt",
    .color = "yellow_black",
    .width = 192,
    .height = 64
}
```

5.1.3.15 map_menu

Assets Arcade::map_menu

Initial value:

```
{
    .image_path = "assets/image/arcade_menu_sprite.png",
    .character = "assets/text/menu.txt",
    .color = "red_black",
    .width = 992,
    .height = 192
}
```

5.1.3.16 snake_body_horizontal

Assets Arcade::snake_body_horizontal

Initial value:

```
{
    .image_path = "assets/image/snake_body_x.png",
    .character = "assets/text/snake_body_x.txt",
    .color = "red_green",
    .width = 32,
    .height = 32
}
```

5.1.3.17 snake_body_vertical

Assets Arcade::snake_body_vertical

Initial value:

```
{
    .image_path = "assets/image/snake_body_y.png",
    .character = "assets/text/snake_body_y.txt",
    .color = "red_green",
    .width = 32,
    .height = 32
}
```

5.1.3.18 snake_head_east

Assets Arcade::snake_head_east

Initial value:

```
{
    .image_path = "assets/image/snake_head_east.png",
    .character = "assets/text/snake_head_east.txt",
    .color = "red_green",
    .width = 32,
    .height = 32
}
```

5.1.3.19 snake_head_north

Assets Arcade::snake_head_north

Initial value:

```
{
    .image_path = "assets/image/snake_head_north.png",
    .character = "assets/text/snake_head_north.txt",
    .color = "red_green",
    .width = 32,
    .height = 32
}
```

5.1.3.20 snake_head_south

Assets Arcade::snake_head_south

Initial value:

```
{
    .image_path = "assets/image/snake_head_south.png",
    .character = "assets/text/snake_head_south.txt",
    .color = "red_green",
    .width = 32,
    .height = 32
}
```

5.1.3.21 snake_head_west

Assets Arcade::snake_head_west

Initial value:

```
{
    .image_path = "assets/image/snake_head_west.png",
    .character = "assets/text/snake_head_west.txt",
    .color = "red_green",
    .width = 32,
    .height = 32
}
```

5.1.3.22 wall

`Assets` `Arcade::wall`

Initial value:

```
{  
    .image_path = "assets/image/cobble_sprite.png",  
    .character = "assets/text/brick.txt",  
    .color = "blue",  
    .width = 32,  
    .height = 32  
}
```


Chapter 6

Class Documentation

6.1 Arcade::Arc Class Reference

Public Member Functions

- void **run** (const std::string &lib_path, const std::string &game_path)
- void **load_graphical** (const std::string &path)
- void **load_games** (const std::string &path)
- void **init** ()

Public Attributes

- const std::string **menu_path** = "lib/arcade_menu.so"
- std::vector< std::string > **_gamesList** = {"lib/arcade_snake.so", "lib/arcade_nibbler.so"}
- std::vector< std::string > **_graphicsList** = {"lib/arcade_ncurses.so", "lib/arcade_sdl2.so", "lib/arcade_sfml.so"}
- bool **_run**
- std::unique_ptr< IGraphicLib > **_graphical** = nullptr
- std::unique_ptr< IGameLib > **_game** = nullptr
- std::unique_ptr< DLLoader< IGraphicLib > > **_graphicalLibrary** = nullptr
- std::unique_ptr< DLLoader< IGameLib > > **_gameLibrary** = nullptr

The documentation for this class was generated from the following files:

- src/Core/Arcade.hpp
- src/Core/Arcade.cpp

6.2 Assets Struct Reference

Public Attributes

- fs::path **image_path**
- fs::path **character**
- fs::path **color**
- int **width**
- int **height**

The documentation for this struct was generated from the following file:

- src/include/Assets.hpp

6.3 DLoader< T > Class Template Reference

Public Member Functions

- **DLoader** (const std::string filename)
- T * **getInstance** ()
- const std::string & **getPath** () const

The documentation for this class was generated from the following file:

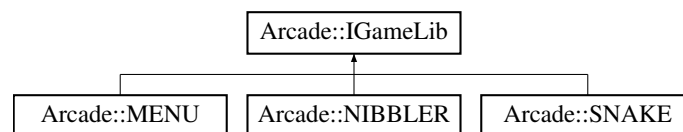
- src/Core/Loader/Loader.hpp

6.4 Arcade::IGameLib Class Reference

Game libraries virtual class.

```
#include <IGameLib.hpp>
```

Inheritance diagram for Arcade::IGameLib:



Public Member Functions

- virtual std::string **getName** () const =0
- virtual bool **init** (Arcade::IGraphicLib &lib)=0
- virtual void **map_creator** (Arcade::IGraphicLib &lib)=0
- virtual void **directions_assigner** (Arcade::Keys key)=0
- virtual void **collisions** ()=0
- virtual void **drawSnake** (Arcade::IGraphicLib &lib, int snake_body_x, int snake_body_y)=0
- virtual void **end_game_score** (Arcade::IGraphicLib &lib)=0
- virtual void **change_libs** (Arcade::IGraphicLib &lib, Arcade::Keys key)=0

6.4.1 Detailed Description

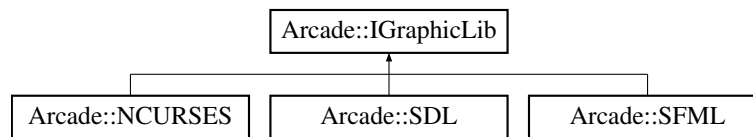
Game libraries virtual class.

The documentation for this class was generated from the following file:

- src/Games/IGameLib.hpp

6.5 Arcade::IGraphicLib Class Reference

Inheritance diagram for Arcade::IGraphicLib:



Public Member Functions

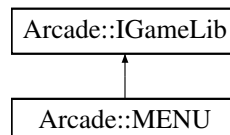
- virtual std::string **getName** () const =0
- virtual void **popWindow** (std::string const &title, int width, int height)=0
- virtual [Keys](#) **getLastEvent** ()=0
- virtual void **mapManager** ([Assets](#) &a, int x, int y)=0
- virtual void **drawText** (std::string text, int x, int y)=0
- virtual void **closeWindow** ()=0
- virtual int **select** ([Arcade::Keys](#) key)=0
- virtual void **refresh** ()=0

The documentation for this class was generated from the following file:

- src/Graphics/IGraphicLib.hpp

6.6 Arcade::MENU Class Reference

Inheritance diagram for Arcade::MENU:



Public Member Functions

- std::string [getName](#) () const override
- bool [init](#) ([Arcade::IGraphicLib](#) &lib) override
- void [directions_assigner](#) ([Arcade::Keys](#) key) override
- void [collisions](#) () override
- void [drawSnake](#) ([Arcade::IGraphicLib](#) &lib, int snake_body_x, int snake_body_y) override
- void [end_game_score](#) ([Arcade::IGraphicLib](#) &lib) override
- void [change_libs](#) ([Arcade::IGraphicLib](#) &lib, [Arcade::Keys](#) key) override
- void [map_creator](#) ([Arcade::IGraphicLib](#) &lib) override
- virtual std::string **getName** () const =0
- virtual bool **init** ([Arcade::IGraphicLib](#) &lib)=0
- virtual void **map_creator** ([Arcade::IGraphicLib](#) &lib)=0
- virtual void **directions_assigner** ([Arcade::Keys](#) key)=0
- virtual void **collisions** ()=0
- virtual void **drawSnake** ([Arcade::IGraphicLib](#) &lib, int snake_body_x, int snake_body_y)=0
- virtual void **end_game_score** ([Arcade::IGraphicLib](#) &lib)=0
- virtual void **change_libs** ([Arcade::IGraphicLib](#) &lib, [Arcade::Keys](#) key)=0

Public Attributes

- int **counter**

6.6.1 Member Function Documentation

6.6.1.1 change_libs()

```
void Arcade::MENU::change_libs (
    Arcade::IGraphicLib & lib,
    Arcade::Keys key ) [override], [virtual]
```

Implements [Arcade::IGameLib](#).

6.6.1.2 collisions()

```
void Arcade::MENU::collisions ( ) [override], [virtual]
```

Implements [Arcade::IGameLib](#).

6.6.1.3 directions_assigner()

```
void Arcade::MENU::directions_assigner (
    Arcade::Keys key ) [override], [virtual]
```

Implements [Arcade::IGameLib](#).

6.6.1.4 drawSnake()

```
void Arcade::MENU::drawSnake (
    Arcade::IGraphicLib & lib,
    int snake_body_x,
    int snake_body_y ) [override], [virtual]
```

Implements [Arcade::IGameLib](#).

6.6.1.5 end_game_score()

```
void Arcade::MENU::end_game_score (
    Arcade::IGraphicLib & lib ) [override], [virtual]
```

Implements [Arcade::IGameLib](#).

6.6.1.6 getName()

```
std::string Arcade::MENU::getName ( ) const [override], [virtual]
```

Implements [Arcade::IGameLib](#).

6.6.1.7 init()

```
bool Arcade::MENU::init (
    Arcade::IGraphicLib & lib ) [override], [virtual]
```

Implements [Arcade::IGameLib](#).

6.6.1.8 map_creator()

```
void Arcade::MENU::map_creator (
    Arcade::IGraphicLib & lib ) [override], [virtual]
```

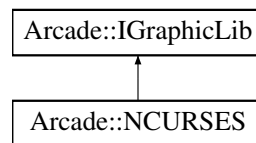
Implements [Arcade::IGameLib](#).

The documentation for this class was generated from the following files:

- src/Games/Menu/Menu.hpp
- src/Games/Menu/Menu.cpp

6.7 Arcade::NCURSES Class Reference

Inheritance diagram for Arcade::NCURSES:



Public Member Functions

- std::string [getName](#) () const override
- void [popWindow](#) (std::string const &title, int width, int height) override
- [Keys](#) [getLastEvent](#) () override
- void [mapManager](#) ([Assets](#) &a, int x, int y) override
- void [drawText](#) (std::string text, int x, int y) override
- void [closeWindow](#) () override
- int [select](#) ([Arcade::Keys](#) key) override
- void [printFile](#) ([Assets](#) &a, fs::path file_path, int x, int y)
- void [refresh](#) () override
- virtual std::string **getName** () const =0
- virtual void **popWindow** (std::string const &title, int width, int height)=0
- virtual [Keys](#) **getLastEvent** ()=0
- virtual void **mapManager** ([Assets](#) &a, int x, int y)=0
- virtual void **drawText** (std::string text, int x, int y)=0
- virtual void **closeWindow** ()=0
- virtual int **select** ([Arcade::Keys](#) key)=0
- virtual void **refresh** ()=0

Public Attributes

- int **lib_count**
- std::string **win_title**

Static Public Attributes

- static WINDOW * **win** = nullptr

6.7.1 Member Function Documentation

6.7.1.1 closeWindow()

```
void Arcade::NCURSES::closeWindow ( ) [override], [virtual]
```

Implements [Arcade::IGraphicLib](#).

6.7.1.2 drawText()

```
void Arcade::NCURSES::drawText (
    std::string text,
    int x,
    int y ) [override], [virtual]
```

Implements [Arcade::IGraphicLib](#).

6.7.1.3 getLastEvent()

```
Keys Arcade::NCURSES::getLastEvent ( ) [override], [virtual]
```

Implements [Arcade::IGraphicLib](#).

6.7.1.4 getName()

```
std::string Arcade::NCURSES::getName ( ) const [override], [virtual]
```

Implements [Arcade::IGraphicLib](#).

6.7.1.5 mapManager()

```
void Arcade::NCURSES::mapManager (
    Assets & a,
    int x,
    int y ) [override], [virtual]
```

Implements [Arcade::IGraphicLib](#).

6.7.1.6 popWindow()

```
void Arcade::NCURSES::popWindow (
    std::string const & title,
    int width,
    int height ) [override], [virtual]
```

Implements [Arcade::IGraphicLib](#).

6.7.1.7 refresh()

```
void Arcade::NCURSES::refresh ( ) [override], [virtual]
```

Implements [Arcade::IGraphicLib](#).

6.7.1.8 select()

```
int Arcade::NCURSES::select (
    Arcade::Keys key ) [override], [virtual]
```

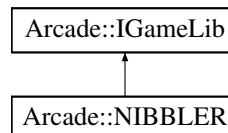
Implements [Arcade::IGraphicLib](#).

The documentation for this class was generated from the following files:

- src/Graphics/NCURSES/NCurses.hpp
- src/Graphics/NCURSES/NCurses.cpp

6.8 Arcade::NIBBLER Class Reference

Inheritance diagram for Arcade::NIBBLER:



Public Member Functions

- std::string [getName](#) () const override
- void [map_creator](#) (Arcade::IGraphicLib &lib) override
- void [directions_assigner](#) (Arcade::Keys key) override
- void [collisions](#) () override
- void [drawSnake](#) (Arcade::IGraphicLib &lib, int snake_body_x, int snake_body_y) override
- void [end_game_score](#) (Arcade::IGraphicLib &lib) override
- void [change_libs](#) (Arcade::IGraphicLib &lib, Arcade::Keys key) override
- bool [init](#) (Arcade::IGraphicLib &lib) override
- virtual std::string **getName** () const =0
- virtual bool **init** (Arcade::IGraphicLib &lib)=0
- virtual void **map_creator** (Arcade::IGraphicLib &lib)=0
- virtual void **directions_assigner** (Arcade::Keys key)=0
- virtual void **collisions** ()=0
- virtual void **drawSnake** (Arcade::IGraphicLib &lib, int snake_body_x, int snake_body_y)=0
- virtual void **end_game_score** (Arcade::IGraphicLib &lib)=0
- virtual void **change_libs** (Arcade::IGraphicLib &lib, Arcade::Keys key)=0

Public Attributes

- int **snake_x** = 1
- int **snake_y** = 1
- int **snake_body_x**
- int **snake_body_y**
- int **food_counter** = 0
- bool **north**
- bool **south**
- bool **east** = true
- bool **west**

6.8.1 Member Function Documentation

6.8.1.1 change_libs()

```
void Arcade::NIBBLER::change_libs (
    Arcade::IGraphicLib & lib,
    Arcade::Keys key ) [override], [virtual]
```

Implements [Arcade::IGameLib](#).

6.8.1.2 collisions()

```
void Arcade::NIBBLER::collisions ( ) [override], [virtual]
```

Implements [Arcade::IGameLib](#).

6.8.1.3 directions_assigner()

```
void Arcade::NIBBLER::directions_assigner (
    Arcade::Keys key ) [override], [virtual]
```

Implements [Arcade::IGameLib](#).

6.8.1.4 drawSnake()

```
void Arcade::NIBBLER::drawSnake (
    Arcade::IGraphicLib & lib,
    int snake_body_x,
    int snake_body_y ) [override], [virtual]
```

Implements [Arcade::IGameLib](#).

6.8.1.5 end_game_score()

```
void Arcade::NIBBLER::end_game_score (
    Arcade::IGraphicLib & lib ) [override], [virtual]
```

Implements [Arcade::IGameLib](#).

6.8.1.6 getName()

```
std::string Arcade::NIBBLER::getName ( ) const [override], [virtual]
```

Implements [Arcade::IGameLib](#).

6.8.1.7 init()

```
bool Arcade::NIBBLER::init (
    Arcade::IGraphicLib & lib ) [override], [virtual]
```

Implements [Arcade::IGameLib](#).

6.8.1.8 map_creator()

```
void Arcade::NIBBLER::map_creator (
    Arcade::IGraphicLib & lib ) [override], [virtual]
```

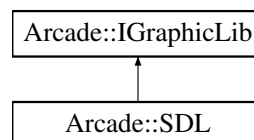
Implements [Arcade::IGameLib](#).

The documentation for this class was generated from the following files:

- src/Games/Nibbler/Nibbler.hpp
- src/Games/Nibbler/Nibbler.cpp

6.9 Arcade::SDL Class Reference

Inheritance diagram for Arcade::SDL:



Public Member Functions

- std::string [getName](#) () const override
- void [popWindow](#) (std::string const &title, int width, int height) override
- [Keys](#) [getLastEvent](#) () override
- void [mapManager](#) ([Assets](#) &a, int x, int y) override
- void [drawText](#) (std::string text, int x, int y) override
- void [closeWindow](#) () override
- int [select](#) ([Arcade::Keys](#) key) override
- void [refresh](#) () override
- virtual std::string [getName](#) () const =0
- virtual void [popWindow](#) (std::string const &title, int width, int height)=0
- virtual [Keys](#) [getLastEvent](#) ()=0
- virtual void [mapManager](#) ([Assets](#) &a, int x, int y)=0
- virtual void [drawText](#) (std::string text, int x, int y)=0
- virtual void [closeWindow](#) ()=0
- virtual int [select](#) ([Arcade::Keys](#) key)=0
- virtual void [refresh](#) ()=0

Public Attributes

- `SDL_Window *` **window**
- `SDL_Surface *` **window_surface**
- `SDL_Renderer *` **renderer**
- `int` **lib_count**
- `bool` **keep_window_open** = true
- `int` **scale_factorx**
- `int` **scale_factory**

6.9.1 Member Function Documentation

6.9.1.1 closeWindow()

```
void Arcade::SDL::closeWindow ( ) [override], [virtual]
```

Implements [Arcade::IGraphicLib](#).

6.9.1.2 drawText()

```
void Arcade::SDL::drawText (
    std::string text,
    int x,
    int y ) [override], [virtual]
```

Implements [Arcade::IGraphicLib](#).

6.9.1.3 getLastEvent()

```
Keys Arcade::SDL::getLastEvent ( ) [override], [virtual]
```

Implements [Arcade::IGraphicLib](#).

6.9.1.4 getName()

```
std::string Arcade::SDL::getName ( ) const [override], [virtual]
```

Implements [Arcade::IGraphicLib](#).

6.9.1.5 mapManager()

```
void Arcade::SDL::mapManager (
    Assets & a,
    int x,
    int y ) [override], [virtual]
```

Implements [Arcade::IGraphicLib](#).

6.9.1.6 popWindow()

```
void Arcade::SDL::popWindow (
    std::string const & title,
    int width,
    int height ) [override], [virtual]
```

Implements [Arcade::IGraphicLib](#).

6.9.1.7 refresh()

```
void Arcade::SDL::refresh ( ) [override], [virtual]
```

Implements [Arcade::IGraphicLib](#).

6.9.1.8 select()

```
int Arcade::SDL::select (
    Arcade::Keys key ) [override], [virtual]
```

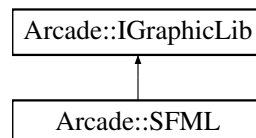
Implements [Arcade::IGraphicLib](#).

The documentation for this class was generated from the following files:

- src/Graphics/SDL/SDL.hpp
- src/Graphics/SDL/SDL.cpp

6.10 Arcade::SFML Class Reference

Inheritance diagram for Arcade::SFML:



Public Member Functions

- std::string [getName](#) () const override
- void [popWindow](#) (std::string const &title, int width, int height) override
- [Keys](#) [getLastEvent](#) () override
- void [mapManager](#) ([Assets](#) &a, int x, int y) override
- void [drawText](#) (std::string text, int x, int y) override
- void [closeWindow](#) () override
- int [select](#) ([Arcade::Keys](#) key) override
- void [refresh](#) () override
- virtual std::string **getName** () const =0
- virtual void **popWindow** (std::string const &title, int width, int height)=0
- virtual [Keys](#) **getLastEvent** ()=0
- virtual void **mapManager** ([Assets](#) &a, int x, int y)=0
- virtual void **drawText** (std::string text, int x, int y)=0
- virtual void **closeWindow** ()=0
- virtual int **select** ([Arcade::Keys](#) key)=0
- virtual void **refresh** ()=0

Public Attributes

- sf::RenderWindow **window**
- sf::VideoMode **videoMode**
- int **lib_count**

6.10.1 Member Function Documentation

6.10.1.1 closeWindow()

```
void Arcade::SFML::closeWindow ( ) [override], [virtual]
```

Implements [Arcade::IGraphicLib](#).

6.10.1.2 drawText()

```
void Arcade::SFML::drawText (
    std::string text,
    int x,
    int y ) [override], [virtual]
```

Implements [Arcade::IGraphicLib](#).

6.10.1.3 getLastEvent()

```
Keys Arcade::SFML::getLastEvent ( ) [override], [virtual]
```

Implements [Arcade::IGraphicLib](#).

6.10.1.4 getName()

```
std::string Arcade::SFML::getName ( ) const [override], [virtual]
```

Implements [Arcade::IGraphicLib](#).

6.10.1.5 mapManager()

```
void Arcade::SFML::mapManager (
    Assets & a,
    int x,
    int y ) [override], [virtual]
```

Implements [Arcade::IGraphicLib](#).

6.10.1.6 popWindow()

```
void Arcade::SFML::popWindow (
    std::string const & title,
    int width,
    int height ) [override], [virtual]
```

Implements [Arcade::IGraphicLib](#).

6.10.1.7 refresh()

```
void Arcade::SFML::refresh ( ) [override], [virtual]
```

Implements [Arcade::IGraphicLib](#).

6.10.1.8 select()

```
int Arcade::SFML::select (
    Arcade::Keys key ) [override], [virtual]
```

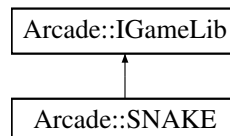
Implements [Arcade::IGraphicLib](#).

The documentation for this class was generated from the following files:

- src/Graphics/SFML/SFML.hpp
- src/Graphics/SFML/SFML.cpp

6.11 Arcade::SNAKE Class Reference

Inheritance diagram for Arcade::SNAKE:



Public Member Functions

- std::string [getName](#) () const override
- void [map_creator](#) (Arcade::IGraphicLib &lib) override
- void [directions_assigner](#) (Arcade::Keys key) override
- void [collisions](#) () override
- void [drawSnake](#) (Arcade::IGraphicLib &lib, int snake_body_x, int snake_body_y) override
- void [end_game_score](#) (Arcade::IGraphicLib &lib) override
- void [change_libs](#) (Arcade::IGraphicLib &lib, Arcade::Keys key) override
- bool [init](#) (Arcade::IGraphicLib &lib) override
- virtual std::string **getName** () const =0
- virtual bool **init** (Arcade::IGraphicLib &lib)=0
- virtual void **map_creator** (Arcade::IGraphicLib &lib)=0
- virtual void **directions_assigner** (Arcade::Keys key)=0
- virtual void **collisions** ()=0
- virtual void **drawSnake** (Arcade::IGraphicLib &lib, int snake_body_x, int snake_body_y)=0
- virtual void **end_game_score** (Arcade::IGraphicLib &lib)=0
- virtual void **change_libs** (Arcade::IGraphicLib &lib, Arcade::Keys key)=0

Public Attributes

- int **snake_x** = 4
- int **snake_y** = 5
- int **snake_body_x**
- int **snake_body_y**
- int **food_counter** = 0
- bool **north**
- bool **south**
- bool **east** = true
- bool **west**

6.11.1 Member Function Documentation

6.11.1.1 change_libs()

```
void Arcade::SNAKE::change_libs (
    Arcade::IGraphicLib & lib,
    Arcade::Keys key ) [override], [virtual]
```

Implements [Arcade::IGameLib](#).

6.11.1.2 collisions()

```
void Arcade::SNAKE::collisions ( ) [override], [virtual]
```

Implements [Arcade::IGameLib](#).

6.11.1.3 directions_assigner()

```
void Arcade::SNAKE::directions_assigner (
    Arcade::Keys key ) [override], [virtual]
```

Implements [Arcade::IGameLib](#).

6.11.1.4 drawSnake()

```
void Arcade::SNAKE::drawSnake (
    Arcade::IGraphicLib & lib,
    int snake_body_x,
    int snake_body_y ) [override], [virtual]
```

Implements [Arcade::IGameLib](#).

6.11.1.5 end_game_score()

```
void Arcade::SNAKE::end_game_score (
    Arcade::IGraphicLib & lib ) [override], [virtual]
```

Implements [Arcade::IGameLib](#).

6.11.1.6 getName()

```
std::string Arcade::SNAKE::getName ( ) const [override], [virtual]
```

Implements [Arcade::IGameLib](#).

6.11.1.7 init()

```
bool Arcade::SNAKE::init (
    Arcade::IGraphicLib & lib ) [override], [virtual]
```

Implements [Arcade::IGameLib](#).

6.11.1.8 map_creator()

```
void Arcade::SNAKE::map_creator (
    Arcade::IGraphicLib & lib ) [override], [virtual]
```

Implements [Arcade::IGameLib](#).

The documentation for this class was generated from the following files:

- src/Games/Snake/Snake.hpp
- src/Games/Snake/Snake.cpp

Chapter 7

File Documentation

7.1 Arcade.cpp

```
00001 /*
00002 ** EPITECH PROJECT, 2023
00003 ** B-OOP-400-LYN-4-1-arcade-mathis.legrand
00004 ** File description:
00005 ** Arcade
00006 */
00007
00008 #include "Arcade.hpp"
00009
00010 void Arcade::Arc::run(const std::string &lib_path, const std::string &game_path)
00011 {
00012     load_graphical(lib_path);
00013     load_games(game_path);
00014
00015     auto clock = std::chrono::steady_clock::now();
00016
00017     _run = true;
00018     while (_run == true) {
00019         if (_run == false)
00020             break;
00021         auto now = std::chrono::steady_clock::now();
00022         auto elapsed = now - clock;
00023         clock = now;
00024
00025         init();
00026     }
00027 }
00028
00029 void Arcade::Arc::init()
00030 {
00031     _game->init(*_graphical);
00032 }
00033
00034 void Arcade::Arc::load_graphical(const std::string &path)
00035 {
00036     std::string copy_path = path;
00037     try {
00038         _graphicalLibrary = std::make_unique<DLloader<IGraphicLib>>(copy_path);
00039         _graphical = std::unique_ptr<Arcade::IGraphicLib>(_graphicalLibrary->getInstance());
00040     }
00041     catch (const std::exception& e) {
00042         std::cerr << e.what() << std::endl;
00043     }
00044 }
00045
00046 void Arcade::Arc::load_games(const std::string &path)
00047 {
00048     std::string copy_path = path;
00049     try {
00050         _gameLibrary = std::make_unique<DLloader<IGameLib>>(copy_path);
00051         _game.reset(_gameLibrary->getInstance());
00052     }
00053     catch (const std::exception& e) {
00054         std::cerr << e.what() << std::endl;
00055     }
00056 }
00057
00058 Arcade::Arc::~Arc()
```

```

00059 {
00060     _game = nullptr;
00061     _graphical = nullptr;
00062 }

```

7.2 Arcade.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2023
00003 ** B-OOP-400-LYN-4-1-arcade-mathis.legrand
00004 ** File description:
00005 ** Arcade
00006 */
00007
00008 #pragma once
00009 #include "Loader/Loader.hpp"
00010 #include "../Graphics/IGraphicLib.hpp"
00011 #include "../Games/IGameLib.hpp"
00012 #include "../include/Assets.hpp"
00013
00014 #include <memory>
00015 #include <string>
00016 #include <vector>
00017 #include <iostream>
00018 #include <string.h>
00019
00020 namespace Arcade {
00021     class Arc {
00022     public:
00023         Arc() = default;
00024         ~Arc();
00025         void run(const std::string &lib_path, const std::string &game_path);
00026         void load_graphical(const std::string &path);
00027         void load_games(const std::string &path);
00028         const std::string menu_path = "lib/arcade_menu.so";
00029         std::vector<std::string> _gamesList = {"lib/arcade_snake.so", "lib/arcade_nibbler.so"};
00030         std::vector<std::string> _graphicalsList = {"lib/arcade_ncurses.so", "lib/arcade_sdl2.so",
00031 "lib/arcade_sfml.so"};
00032         void init();
00033         bool _run;
00034
00035         std::unique_ptr<IGraphicLib> _graphical = nullptr;
00036         std::unique_ptr<IGameLib> _game = nullptr;
00037         std::unique_ptr<DLLoader<IGraphicLib> _graphicalLibrary = nullptr;
00038         std::unique_ptr<DLLoader<IGameLib> _gameLibrary = nullptr;
00039     private:
00040     };
00041 }

```

7.3 Loader.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2023
00003 ** B-OOP-400-LYN-4-1-arcade-mathis.legrand
00004 ** File description:
00005 ** Loader
00006 */
00007
00008 #pragma once
00009
00010 #include <iostream>
00011 #include <string>
00012 #include <vector>
00013 #include <dlfcn.h>
00014
00015 template<typename T>
00016 class DLoader {
00017 private:
00018     std::string _path;
00019     void *handle;
00020     T *(*get_instance)();
00021
00022 public:
00023     DLoader(const std::string filename)
00024     {
00025         handle = dlopen(filename.c_str(), RTLD_LAZY);
00026         if (!handle) {
00027             throw std::runtime_error(dlerror());

```



```

00028         }
00029         get_instance = reinterpret_cast<T * (*) ()>(dlsym(handle, "getInstance"));
00030         if (!get_instance) {
00031             throw std::runtime_error(dlerror());
00032         }
00033     }
00034     ~DLLoader()
00035     {
00036         if (handle)
00037         {
00038             dlclose(handle);
00039         }
00040     }
00041 }
00042
00043 T *getInstance()
00044 {
00045     return get_instance();
00046 }
00047
00048 const std::string &getPath() const
00049 {
00050     return _path;
00051 }
00052 };

```

7.4 IGameLib.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2023
00003 ** arcade
00004 ** File description:
00005 ** arcade
00006 */
00007
00008 /*
00009 * Interface used by game libraries.
00010 * All functions must be implemented correctly for the kernel to handle the game libraries.
00011 */
00012
00013 #pragma once
00014
00015 #include "../Graphics/IGraphicLib.hpp"
00016 #include "../include/Assets.hpp"
00017 #include <string>
00018 #include <random>
00019 #include <iostream>
00020 #include <fstream>
00021 #include <chrono>
00022 #include <thread>
00023 #include <cstdlib> // for rand() and srand()
00024 #include <ctime>   // for time()
00025
00026 namespace Arcade {
00027     class IGameLib {
00028     public:
00029         virtual ~IGameLib() = default;
00030         // Destructor
00031         // IGameLib class's destructor
00032
00033         virtual std::string getName() const = 0;
00034         //summary: Game name's getter
00035         //return: a string containing the name of the game
00036         virtual bool init(Arcade::IGraphicLib &lib) = 0;
00037         //summary: Initializes the game
00038         //param lib: the graphic library to use
00039         //return: true if the game was initialized correctly, false otherwise
00040
00041         virtual void map_creator(Arcade::IGraphicLib &lib) = 0;
00042         //summary: Creates the map
00043         //param lib: the graphic library to use
00044
00045         virtual void directions_assigner(Arcade::Keys key) = 0;
00046         //summary: Assigns the directions to the snake
00047         //param key: the key pressed by the user
00048
00049         virtual void collisions() = 0;
00050         //summary: Checks for collisions
00051
00052         virtual void drawSnake(Arcade::IGraphicLib &lib, int snake_body_x, int snake_body_y) = 0;
00053         //summary: Draws the snake
00054     };
00055 }

```

```

00061         //param lib: the graphic library to use and the snake's body's coordinates
00062
00063         virtual void end_game_score(Arcade::IGraphicLib &lib) = 0;
00064         //summary: Displays the score at the end of the game
00065         //param lib: the graphic library to use
00066
00067         virtual void change_libs(Arcade::IGraphicLib &lib, Arcade::Keys key) = 0;
00068         //summary: Changes the .so file of the graphic or game libraries
00069         //param lib: the graphical interface to use
00070
00071     };
00072 };

```

7.5 Menu.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2023
00003 ** menu header file
00004 ** File description:
00005 ** Menu
00006 */
00007
00008 #pragma once
00009 #include "../IGameLib.hpp"
00010 #include "../../Graphics/IGraphicLib.hpp"
00011
00012 namespace Arcade {
00013
00014     class MENU : public IGameLib {
00015     public:
00016         MENU();
00017         ~MENU();
00018
00019         int counter;
00020         std::string getName() const override;
00021         bool init(Arcade::IGraphicLib &lib) override;
00022         void directions_assigner(Arcade::Keys key) override;
00023         void collisions() override;
00024         void drawSnake(Arcade::IGraphicLib &lib, int snake_body_x, int snake_body_y) override;
00025         void end_game_score(Arcade::IGraphicLib &lib) override;
00026         void change_libs(Arcade::IGraphicLib &lib, Arcade::Keys key) override;
00027         void map_creator(Arcade::IGraphicLib &lib) override;
00028
00029     protected:
00030     private:
00031     };
00032 }

```

7.6 Nibbler.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2023
00003 ** snake header file
00004 ** File description:
00005 ** Nibbler
00006 */
00007
00008 #pragma once
00009 #include "../IGameLib.hpp"
00010 #include "../../Graphics/IGraphicLib.hpp"
00011
00012 namespace Arcade {
00013
00014     class NIBBLER : public IGameLib {
00015     public:
00016         NIBBLER();
00017         ~NIBBLER();
00018
00019         int snake_x = 1;
00020         int snake_y = 1;
00021         int snake_body_x;
00022         int snake_body_y;
00023         int food_counter = 0;
00024         bool north;
00025         bool south;
00026         bool east = true;
00027         bool west;
00028         std::string getName() const override;
00029         void map_creator(Arcade::IGraphicLib &lib) override;

```

```

00030         void directions_assigner(Arcade::Keys key) override;
00031         void collisions() override;
00032         void drawSnake(Arcade::IGraphicLib &lib, int snake_body_x, int snake_body_y) override;
00033         void end_game_score(Arcade::IGraphicLib &lib) override;
00034         void change_libs(Arcade::IGraphicLib &lib, Arcade::Keys key) override;
00035         bool init(Arcade::IGraphicLib &lib) override;
00036
00037     protected:
00038     private:
00039         enum Grid {
00040             Empty,
00041             E = Empty,
00042             Wall,
00043             W = Wall,
00044             Food,
00045             F = Food,
00046             Snake,
00047             S = Snake
00048         };
00049         std::array<Grid, 19*19> _grid = {
00050             W, W, W, W, W, W, W, W, W, W, W, W, W, W, W, W, W, W, W,
00051             W, E, E, F, E, E, E, F, E, E, E, F, E, E, E, F, E, E, W,
00052             W, E, W, W, W, E, W, E, W, W, W, E, W, E, W, W, W, E, W,
00053             W, F, W, E, W, F, W, E, E, E, E, E, W, F, W, E, W, F, W,
00054             W, E, W, W, W, E, W, E, W, E, W, E, W, E, W, W, W, E, W,
00055             W, E, E, E, E, E, W, F, W, E, W, F, W, E, E, E, E, E, W,
00056             W, E, W, W, W, W, W, E, W, E, W, E, W, W, W, W, W, E, W,
00057             W, F, W, E, F, E, E, E, E, E, E, E, E, F, E, W, F, W,
00058             W, E, W, E, W, W, W, W, W, F, W, W, W, W, W, E, W, E, W,
00059             W, E, E, E, F, E, E, E, E, E, E, E, E, E, F, E, E, E, W,
00060             W, E, W, W, W, W, W, E, W, W, W, E, W, W, W, W, W, E, W,
00061             W, E, E, F, E, E, E, E, W, E, W, E, E, E, E, F, E, E, W,
00062             W, E, W, W, W, E, W, E, W, W, W, E, W, E, W, W, W, E, W,
00063             W, E, W, E, W, E, W, E, E, F, E, E, W, E, W, E, W, E, W,
00064             W, F, W, W, W, F, W, F, W, W, W, F, W, F, W, W, W, F, W,
00065             W, E, E, E, E, E, W, E, W, E, W, E, W, E, E, E, E, W,
00066             W, F, W, W, W, W, W, E, W, W, W, E, W, W, W, W, W, F, W,
00067             W, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, W,
00068             W, W, W, W, W, W, W, W, W, W, W, W, W, W, W, W, W, W,
00069         };
00070     };
00071 }

```

7.7 Snake.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2023
00003 ** snake header file
00004 ** File description:
00005 ** Snake
00006 */
00007
00008 #pragma once
00009 #include "../IGameLib.hpp"
00010
00011 namespace Arcade {
00012
00013     class SNAKE : public IGameLib {
00014     public:
00015         SNAKE();
00016         ~SNAKE();
00017
00018         int snake_x = 4;
00019         int snake_y = 5;
00020         int snake_body_x;
00021         int snake_body_y;
00022         int food_counter = 0;
00023         bool north;
00024         bool south;
00025         bool east = true;
00026         bool west;
00027         std::string getName() const override;
00028         void map_creator(Arcade::IGraphicLib &lib) override;
00029         void directions_assigner(Arcade::Keys key) override;
00030         void collisions() override;
00031         void drawSnake(Arcade::IGraphicLib &lib, int snake_body_x, int snake_body_y) override;
00032         void end_game_score(Arcade::IGraphicLib &lib) override;
00033         void change_libs(Arcade::IGraphicLib &lib, Arcade::Keys key) override;
00034         bool init(Arcade::IGraphicLib &lib) override;
00035
00036     protected:
00037     private:
00038         enum Grid {

```

```

00039         Empty,
00040         E = Empty,
00041         Wall,
00042         W = Wall,
00043         Snake,
00044         S = Snake,
00045         Food,
00046         F = Food
00047     };
00048     std::array<Grid, 19*19> _grid = {
00049         W, W, W, W, W, W, W, W, W, W, W, W, W, W, W, W, W, W, W,
00050         W, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, W,
00051         W, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, W,
00052         W, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, W,
00053         W, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, W,
00054         W, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, W,
00055         W, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, W,
00056         W, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, W,
00057         W, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, W,
00058         W, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, W,
00059         W, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, W,
00060         W, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, W,
00061         W, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, W,
00062         W, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, W,
00063         W, E, E, E, E, E, E, E, E, E, E, E, E, E, F, E, E, E, E, W,
00064         W, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, W,
00065         W, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, W,
00066         W, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, E, W,
00067         W, W, W, W, W, W, W, W, W, W, W, W, W, W, W, W, W, W, W,
00068     };
00069 };
00070 }

```

7.8 IGraphicLib.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2023
00003 ** arcade
00004 ** File description:
00005 ** arcade
00006 */
00007
00008 /*
00009 * Interface used by graphic libraries
00010 * All functions must be implemented correctly for the kernel to handle the graphic libraries.
00011 */
00012
00013 #pragma once
00014
00015 #include <string>
00016 #include "../include/Keys.hpp"
00017 #include "../include/Assets.hpp"
00018
00019 namespace Arcade {
00020     class IGraphicLib {
00021     public:
00022         //Purely virtual class that serves as the basis for all graphic libraries
00023
00024         virtual ~IGraphicLib() = default;
00025         // Destructor
00026         //IGraphicLib class's destructor
00027
00028         virtual std::string getName() const = 0;
00029         //summary: Graphic library name's getter
00030         //return: a string containing the name of the graphic library
00031
00032         virtual void popWindow(std::string const& title, int width, int height) = 0;
00033         //summary: Creates a window
00034         //param title: the title of the window
00035         //return: void
00036
00037         virtual Keys getLastEvent() = 0;
00038         //summary: Gets the last keyboard event that happened in the window
00039         //return: the last event that happened in the window
00040
00041         virtual void mapManager(Assets &a, int x, int y) = 0;
00042         //summary: displays different sprites at the x and y position
00043         //param a: the assets to use
00044         //param x: the x position of the map
00045         //param y: the y position of the map
00046
00047         virtual void drawText(std::string text, int x, int y) = 0;
00048     };
00049 }

```

```

00054         //summary: Draws text
00055         //param lib:the text to draw, the x and y coordinates
00056
00057         virtual void closeWindow() = 0;
00058         //summary: Closes the window
00059
00060         virtual int select(Arcade::Keys key) = 0;
00061         //summary: Selects the current item in the menu
00062
00063         virtual void refresh() = 0;
00064         //summary: Refreshes the window
00065     };
00066 };

```

7.9 NCurses.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2023
00003 ** ncurses header file
00004 ** File description:
00005 ** ncurses
00006 */
00007
00008 #pragma once
00009
00010 #include "../IGraphicLib.hpp"
00011 #include <ncurses.h>
00012
00013 //to compile, use these flags: -lncurses
00014
00015 namespace Arcade {
00016
00017     class NCURSES : public IGraphicLib {
00018     public:
00019         NCURSES();
00020         ~NCURSES() override;
00021
00022         static WINDOW* win;
00023         int lib_count;
00024         std::string win_title;
00025         std::string getName() const override;
00026         void popWindow(std::string const& title, int width, int height) override;
00027         Keys getLastEvent() override;
00028         void mapManager(Assets &a, int x, int y) override;
00029         void drawText(std::string text, int x, int y) override;
00030         void closeWindow() override;
00031         int select(Arcade::Keys key) override;
00032         void printFile(Assets &a, fs::path file_path, int x, int y);
00033         void refresh() override;
00034
00035     };
00036 }

```

7.10 SDL.hpp

```

00001 #include "../IGraphicLib.hpp"
00002 #include <iostream>
00003 #include <SDL2/SDL.h>
00004 #include <SDL2/SDL_ttf.h>
00005 #include <SDL2/SDL_image.h>
00006
00007 //to compile use these flags: -lSDL2 -lSDL2_ttf
00008
00009 namespace Arcade {
00010
00011     class SDL : public IGraphicLib {
00012     public:
00013         SDL();
00014         ~SDL() override;
00015         SDL_Window *window;
00016         SDL_Surface *window_surface;
00017         SDL_Renderer *renderer;
00018         int lib_count;
00019         bool keep_window_open = true;
00020         int scale_factorx;
00021         int scale_factory;
00022
00023         std::string getName() const override;
00024         void popWindow(std::string const& title, int width, int height) override;

```

```

00025         Keys getLastEvent() override;
00026         void mapManager(Assets &a, int x, int y) override;
00027         void drawText(std::string text, int x, int y) override;
00028         void closeWindow() override;
00029         int select(Arcade::Keys key) override;
00030         void refresh() override;
00031
00032     private:
00033     };
00034 }

```

7.11 SFML.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2023
00003 ** sfml header file
00004 ** File description:
00005 ** sfml
00006 */
00007
00008 #pragma once
00009
00010 #include "../IGraphicLib.hpp"
00011 #include <SFML/Graphics.hpp>
00012
00013 //to compile use these flags: -lsfml-graphics -lsfml-window -lsfml-system
00014
00015 namespace Arcade {
00016
00017     class SFML : public IGraphicLib {
00018     public:
00019         SFML();
00020         ~SFML() override;
00021
00022         sf::RenderWindow window;
00023         sf::VideoMode videoMode;
00024         int lib_count;
00025         std::string getName() const override;
00026         void popWindow(std::string const& title, int width, int height) override;
00027         Keys getLastEvent() override;
00028         void mapManager(Assets &a, int x, int y) override;
00029         void drawText(std::string text, int x, int y) override;
00030         void closeWindow() override;
00031         int select(Arcade::Keys key) override;
00032         void refresh() override;
00033
00034     private:
00035     };
00036 }

```

7.12 Assets.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2023
00003 ** Untitled (Workspace)
00004 ** File description:
00005 ** Assets
00006 */
00007
00008 #ifndef ASSETS_HPP_
00009 #define ASSETS_HPP_
00010
00011 #include <filesystem>
00012 #include <iostream>
00013 #include <fstream>
00014 #include <string>
00015 #include <cstring>
00016 #include <array>
00017
00018 namespace fs = std::filesystem;
00019
00020 struct Assets {
00021     fs::path image_path;
00022     fs::path character;
00023     fs::path color;
00024     int width;
00025     int height;
00026 };
00027
00028 #endif /* !ASSETS_HPP_ */

```

7.13 Keys.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2023
00003 ** arcade
00004 ** File description:
00005 ** arcade
00006 */
00007
00008 /*
00009 * Key Enum, each graphics library must store a map in order to convert
00010 * the specific library key code into one of this enum code so that it can be
00011 * used by other components independently of the graphics library.
00012 */
00013
00014 #pragma once
00015
00016 namespace Arcade {
00017     enum Keys {
00018         NONE,
00019         A, B, C, D, E, F, G, H, I, J, K, L, M,
00020         N, O, P, Q, R, S, T, U, V, W, X, Y, Z,
00021         LEFT, RIGHT, UP, DOWN,
00022         ENTER, SPACE, DELETE, BACKSPACE, TAB, ESC,
00023         MOUSELEFT, MOUSERIGHT
00024     };
00025
00026
00032 };
```


Index

- apple
 - Arcade, [10](#)
- Arcade, [9](#)
 - apple, [10](#)
 - game_nibbler, [10](#)
 - game_nibbler_selected, [11](#)
 - game_snake, [11](#)
 - game_snake_selected, [11](#)
 - game_title, [11](#)
 - grass, [11](#)
 - Keys, [10](#)
 - lib_ncurses, [12](#)
 - lib_ncurses_selected, [12](#)
 - lib_sdl, [12](#)
 - lib_sdl_selected, [12](#)
 - lib_sfml, [12](#)
 - lib_sfml_selected, [13](#)
 - lib_title, [13](#)
 - map_menu, [13](#)
 - snake_body_horizontal, [13](#)
 - snake_body_vertical, [13](#)
 - snake_head_east, [14](#)
 - snake_head_north, [14](#)
 - snake_head_south, [14](#)
 - snake_head_west, [14](#)
 - wall, [14](#)
- Arcade::Arc, [17](#)
- Arcade::IGameLib, [18](#)
- Arcade::IGraphicLib, [19](#)
- Arcade::MENU, [19](#)
 - change_libs, [20](#)
 - collisions, [20](#)
 - directions_assigner, [20](#)
 - drawSnake, [20](#)
 - end_game_score, [20](#)
 - getName, [20](#)
 - init, [20](#)
 - map_creator, [21](#)
- Arcade::NCURSES, [21](#)
 - closeWindow, [22](#)
 - drawText, [22](#)
 - getLastEvent, [22](#)
 - getName, [22](#)
 - mapManager, [22](#)
 - popWindow, [22](#)
 - refresh, [23](#)
 - select, [23](#)
- Arcade::NIBBLER, [23](#)
 - change_libs, [24](#)
 - collisions, [24](#)
 - directions_assigner, [24](#)
 - drawSnake, [24](#)
 - end_game_score, [24](#)
 - getName, [24](#)
 - init, [25](#)
 - map_creator, [25](#)
- Arcade::SDL, [25](#)
 - closeWindow, [26](#)
 - drawText, [26](#)
 - getLastEvent, [26](#)
 - getName, [26](#)
 - mapManager, [26](#)
 - popWindow, [26](#)
 - refresh, [27](#)
 - select, [27](#)
- Arcade::SFML, [27](#)
 - closeWindow, [28](#)
 - drawText, [28](#)
 - getLastEvent, [28](#)
 - getName, [28](#)
 - mapManager, [28](#)
 - popWindow, [28](#)
 - refresh, [29](#)
 - select, [29](#)
- Arcade::SNAKE, [29](#)
 - change_libs, [30](#)
 - collisions, [30](#)
 - directions_assigner, [30](#)
 - drawSnake, [30](#)
 - end_game_score, [30](#)
 - getName, [30](#)
 - init, [31](#)
 - map_creator, [31](#)
- Assets, [17](#)
- change_libs
 - Arcade::MENU, [20](#)
 - Arcade::NIBBLER, [24](#)
 - Arcade::SNAKE, [30](#)
- closeWindow
 - Arcade::NCURSES, [22](#)
 - Arcade::SDL, [26](#)
 - Arcade::SFML, [28](#)
- collisions
 - Arcade::MENU, [20](#)
 - Arcade::NIBBLER, [24](#)
 - Arcade::SNAKE, [30](#)
- directions_assigner

- Arcade::MENU, [20](#)
 - Arcade::NIBBLER, [24](#)
 - Arcade::SNAKE, [30](#)
- DLLoader< T >, [18](#)
- drawSnake
 - Arcade::MENU, [20](#)
 - Arcade::NIBBLER, [24](#)
 - Arcade::SNAKE, [30](#)
- drawText
 - Arcade::NCURSES, [22](#)
 - Arcade::SDL, [26](#)
 - Arcade::SFML, [28](#)
- end_game_score
 - Arcade::MENU, [20](#)
 - Arcade::NIBBLER, [24](#)
 - Arcade::SNAKE, [30](#)
- game_nibbler
 - Arcade, [10](#)
- game_nibbler_selected
 - Arcade, [11](#)
- game_snake
 - Arcade, [11](#)
- game_snake_selected
 - Arcade, [11](#)
- game_title
 - Arcade, [11](#)
- getLastEvent
 - Arcade::NCURSES, [22](#)
 - Arcade::SDL, [26](#)
 - Arcade::SFML, [28](#)
- getName
 - Arcade::MENU, [20](#)
 - Arcade::NCURSES, [22](#)
 - Arcade::NIBBLER, [24](#)
 - Arcade::SDL, [26](#)
 - Arcade::SFML, [28](#)
 - Arcade::SNAKE, [30](#)
- grass
 - Arcade, [11](#)
- init
 - Arcade::MENU, [20](#)
 - Arcade::NIBBLER, [25](#)
 - Arcade::SNAKE, [31](#)
- Keys
 - Arcade, [10](#)
- lib_ncurses
 - Arcade, [12](#)
- lib_ncurses_selected
 - Arcade, [12](#)
- lib_sdl
 - Arcade, [12](#)
- lib_sdl_selected
 - Arcade, [12](#)
- lib_sfml
 - Arcade, [12](#)
- lib_sfml_selected
 - Arcade, [13](#)
- lib_title
 - Arcade, [13](#)
- map_creator
 - Arcade::MENU, [21](#)
 - Arcade::NIBBLER, [25](#)
 - Arcade::SNAKE, [31](#)
- map_menu
 - Arcade, [13](#)
- mapManager
 - Arcade::NCURSES, [22](#)
 - Arcade::SDL, [26](#)
 - Arcade::SFML, [28](#)
- popWindow
 - Arcade::NCURSES, [22](#)
 - Arcade::SDL, [26](#)
 - Arcade::SFML, [28](#)
- refresh
 - Arcade::NCURSES, [23](#)
 - Arcade::SDL, [27](#)
 - Arcade::SFML, [29](#)
- select
 - Arcade::NCURSES, [23](#)
 - Arcade::SDL, [27](#)
 - Arcade::SFML, [29](#)
- snake_body_horizontal
 - Arcade, [13](#)
- snake_body_vertical
 - Arcade, [13](#)
- snake_head_east
 - Arcade, [14](#)
- snake_head_north
 - Arcade, [14](#)
- snake_head_south
 - Arcade, [14](#)
- snake_head_west
 - Arcade, [14](#)
- src/Core/Arcade.cpp, [33](#)
- src/Core/Arcade.hpp, [34](#)
- src/Core/Loader/Loader.hpp, [34](#)
- src/Games/IGameLib.hpp, [35](#)
- src/Games/Menu/Menu.hpp, [36](#)
- src/Games/Nibbler/Nibbler.hpp, [36](#)
- src/Games/Snake/Snake.hpp, [37](#)
- src/Graphics/IGraphicLib.hpp, [38](#)
- src/Graphics/NCURSES/NCurses.hpp, [39](#)
- src/Graphics/SDL/SDL.hpp, [39](#)
- src/Graphics/SFML/SFML.hpp, [40](#)
- src/include/Assets.hpp, [40](#)
- src/include/Keys.hpp, [41](#)
- wall
 - Arcade, [14](#)