

My Project

Generated by Doxygen 1.9.5

Chapter 1

Base Architecture for the Arcade Project

1.1 Interfaces

Since the main goal of the project is to have multiple different libs used through the same interface, there is 2 of them :

- ILib
- IGame

1.2 Build

To build the project:

```
/B-OOP-400> mkdir ./build/ && cd ./build/  
/B-OOP-400/build> cmake .. -G "Unix Makefiles" -DCMAKE_BUILD_TYPE=Release  
[...]  
/B-OOP-400/build> cmake --build .  
[...]  
/B-OOP-400/build> cd ..  
/B-OOP-400> ls ./arcade ./lib/  
./arcade  
./lib/:  
arcade_ncurses.so  
arcade_sdl2.so  
arcade_sfml.so  
arcade_snake.so  
arcade_nibbler.so
```

Then you'll have a makefile build. But not an Epitech one. Use `make clean` instead of the `make fclean`.

You should watch this video to understand cmake : [CMake](#)

By Dlore made up of:

- [Alexandre Franquet](#) | [Miouzora](#)
- [Quentin Brejoin](#) | [Queng](#)
- [Guylain Thomas](#) | [Jibea](#)

In collaboration with C11H17NO3 made up of:

- [Timothee Lesellier](#) | [Big T](#)
- [Lylia Hay](#) | [VyOk](#)
- [Mathis Evain](#) | [Kibatsu](#)

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

| | |
|---|----|
| Arcade::Core | ?? |
| std::exception | |
| Arcade::LoaderException | ?? |
| Arcade::IGame | ?? |
| Arcade::AGame | ?? |
| Arcade::ILib | ?? |
| Arcade::ALib | ?? |
| Arcade::IObject | ?? |
| Arcade::AObject | ?? |
| Arcade::LibLoader< LibInterface > | ?? |
| Arcade::Text | ?? |

Chapter 3

Class Index

3.1 Class List

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

| | |
|---|----|
| Arcade::AGame | |
| Abstract class for the game | ?? |
| Arcade::ALib | |
| Abstract class for the graphical library | ?? |
| Arcade::AObject | |
| Abstract class for every entities in games | ?? |
| Arcade::Core | ?? |
| Arcade::IGame | |
| Interface for one game library | ?? |
| Arcade::ILib | |
| Interface for the graphical library | ?? |
| Arcade::IObject | |
| Interface for the object | ?? |
| Arcade::LibLoader< LibInterface > | ?? |
| Arcade::LoaderException | |
| The class LoaderException is the exception thrown when the library cannot be loaded | ?? |
| Arcade::Text | |
| The class Text is the generic way to handle text | ?? |

Chapter 4

File Index

4.1 File List

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

| | |
|---|----|
| core/ Core.hpp | ?? |
| core/ ErrorManagement.hpp | ?? |
| core/ LibLoader.hpp | ?? |
| core/src/Game/ AGame.hpp | ?? |
| core/src/Game/ IGame.hpp | ?? |
| core/src/Lib/ ALib.hpp | ?? |
| core/src/Lib/ ILib.hpp | ?? |
| core/src/Lib/ Types.hpp | ?? |
| core/src/Object/ AObjects.hpp | ?? |
| core/src/Object/ IObject.hpp | ?? |
| core/src/Text/ Text.hpp | ?? |

Chapter 5

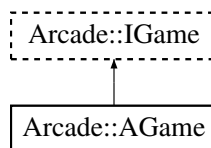
Class Documentation

5.1 Arcade::AGame Class Reference

Abstract class for the game.

```
#include <AGame.hpp>
```

Inheritance diagram for Arcade::AGame:



Public Member Functions

- Arcade::gameState [getState](#) (void) const final override
Get the game state.
- ssize_t [getScore](#) (void) const final override
Get the game score.
- ssize_t [getHighScore](#) (void) const final override
Get the game high score.
- std::string [getGameName](#) (void) const final override
Get the game name.

Protected Attributes

- std::string **_name**
The name of the game.
- Arcade::gameState **_state**
The actual state of the game.
- ssize_t **_score**
The actual score of the game.
- ssize_t **_highScore**
The high score of the game.

5.1.1 Detailed Description

Abstract class for the game.

5.1.2 Member Function Documentation

5.1.2.1 `getGameName()`

```
std::string Arcade::AGame::getGameName (
    void ) const [inline], [final], [override], [virtual]
```

Get the game name.

This function will return the name of the game.

Returns

The name of the game.

Implements [Arcade::IGame](#).

```
24 {return _name;};
```

5.1.2.2 `getHighScore()`

```
ssize_t Arcade::AGame::getHighScore (
    void ) const [inline], [final], [override], [virtual]
```

Get the game high score.

This function will return the current high score of the game.

Returns

The current high score of the game.

Implements [Arcade::IGame](#).

```
23 {return _highScore;};
```

5.1.2.3 getScore()

```
ssize_t Arcade::AGame::getScore (
    void ) const [inline], [final], [override], [virtual]
```

Get the game score.

This function will return the current score of the game.

Returns

The current score of the game.

Implements [Arcade::IGame](#).

```
22 {return _score;};
```

5.1.2.4 getState()

```
Arcade::gameState Arcade::AGame::getState (
    void ) const [inline], [final], [override], [virtual]
```

Get the game state.

This function will return the current state of the game.

Returns

The current state of the game.

Implements [Arcade::IGame](#).

```
21 {return _state;};
```

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

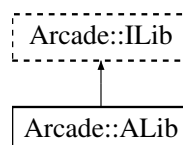
- core/src/Game/AGame.hpp

5.2 Arcade::ALib Class Reference

Abstract class for the graphical library.

```
#include <ALib.hpp>
```

Inheritance diagram for Arcade::ALib:



Public Member Functions

- void [setScale](#) (std::pair< ssize_t, ssize_t > scale) final override
Set the Scale of the window.
- void [setScale](#) (ssize_t scale) final override
Set the Scale of the window.
- void [setSize](#) (std::pair< ssize_t, ssize_t > size) final override
Set the Size of the window.
- std::pair< ssize_t, ssize_t > [getScale](#) (void) const final override
Get the Scale of the window.
- std::pair< ssize_t, ssize_t > [getSize](#) (void) const final override
Get the Size of the window.
- bool [isKeyPressed](#) (Arcade::Inputs) override
Check if a key is pressed.
- bool [isKeyReleased](#) (Arcade::Inputs) override
Check if a key is released.

Protected Attributes

- std::unordered_map< Arcade::Inputs, bool > **_keys**
The map of all key pressed or not.
- std::string **_name**
The name of the library.

5.2.1 Detailed Description

Abstract class for the graphical library.

5.2.2 Member Function Documentation

5.2.2.1 getScale()

```
std::pair< ssize_t, ssize_t > Arcade::ALib::getScale (
    void ) const [inline], [final], [override], [virtual]
```

Get the Scale of the window.

Returns

std::pair<ssize_t, ssize_t>

Implements [Arcade::ILib](#).

```
25 { return _scale; }
```

5.2.2.2 getSize()

```
std::pair< ssize_t, ssize_t > Arcade::ALib::getSize (
    void ) const [inline], [final], [override], [virtual]
```

Get the Size of the window.

Returns

`std::pair<ssize_t, ssize_t>`

Implements [Arcade::ILib](#).

```
26 { return _size; }
```

5.2.2.3 isKeyPressed()

```
bool Arcade::ALib::isKeyPressed (
    Arcade::Inputs input ) [inline], [override], [virtual]
```

Check if a key is pressed.

Parameters

| | |
|--------------|------------------|
| <i>input</i> | The key to check |
|--------------|------------------|

Returns

true if the key is pressed, false otherwise

Implements [Arcade::ILib](#).

```
28 { return false; }
```

5.2.2.4 isKeyReleased()

```
bool Arcade::ALib::isKeyReleased (
    Arcade::Inputs input ) [inline], [override], [virtual]
```

Check if a key is released.

Parameters

| | |
|--------------|------------------|
| <i>input</i> | The key to check |
|--------------|------------------|

Returns

true if the key is released, false otherwise

Implements [Arcade::ILib](#).

```
29 { return false; }
```

5.2.2.5 setScale() [1/2]

```
void Arcade::ALib::setScale (
    ssize_t scale ) [inline], [final], [override], [virtual]
```

Set the Scale of the window.

Parameters

| | |
|--------------|------------------|
| <i>scale</i> | The scale to set |
|--------------|------------------|

Implements [Arcade::ILib](#).

```
23 { _scale = std::pair<ssize_t, ssize_t>(scale, scale); }
```

5.2.2.6 setScale() [2/2]

```
void Arcade::ALib::setScale (
    std::pair< ssize_t, ssize_t > scale ) [inline], [final], [override], [virtual]
```

Set the Scale of the window.

Parameters

| | |
|--------------|------------------|
| <i>scale</i> | The scale to set |
|--------------|------------------|

Implements [Arcade::ILib](#).

```
22 { _scale = scale; }
```

5.2.2.7 setSize()

```
void Arcade::ALib::setSize (
    std::pair< ssize_t, ssize_t > size ) [inline], [final], [override], [virtual]
```

Set the Size of the window.

Parameters

| | |
|-------------|-----------------|
| <i>size</i> | The size to set |
|-------------|-----------------|

Implements [Arcade::ILib](#).

```
24 { _size = size; }
```


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

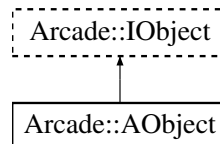
- core/src/Lib/ALib.hpp

5.3 Arcade::AObject Class Reference

Abstract class for every entities in games.

```
#include <AObjects.hpp>
```

Inheritance diagram for Arcade::AObject:



Public Member Functions

- Arcade::Shapes [getShape](#) (void) const final override
Get the Shape object.
- std::pair< ssize_t, ssize_t > [getPosition](#) (void) const final override
Get the Position object.
- std::pair< ssize_t, ssize_t > [getSize](#) (void) const final override
Get the Size object.
- Arcade::Colors [getColor](#) (void) const final override
Set the Color object.
- std::string [getFilePath](#) (void) const final override
Get the Texture object.
- void [setShape](#) (Arcade::Shapes shape) final override
Set the Shape object.
- void [setPosition](#) (std::pair< ssize_t, ssize_t > position) final override
Set the Position object.
- void [setSize](#) (std::pair< ssize_t, ssize_t > size) final override
Set the Size object.
- void [setColor](#) (Arcade::Colors color) final override
Get the Color object.
- void [setFilePath](#) (std::string texture) final override
Set the Texture object.

Protected Attributes

- Arcade::Shapes **_shape**
The shape of the object.
- std::pair< ssize_t, ssize_t > **_position**
The position of the object.
- std::pair< ssize_t, ssize_t > **_size**
The size of the object.
- Arcade::Colors **_color**
The color of the object.
- std::string **_texture**
The texture of the object.

5.3.1 Detailed Description

Abstract class for every entities in games.

5.3.2 Member Function Documentation

5.3.2.1 getColor()

```
Arcade::Colors Arcade::AObject::getColor (
    void ) const [inline], [final], [override], [virtual]
```

Set the Color object.

Parameters

| | |
|--------------|--|
| <i>color</i> | |
|--------------|--|

Implements [Arcade::IObject](#).

```
24 {return _color;;}
```

5.3.2.2 getFilePath()

```
std::string Arcade::AObject::getFilePath (
    void ) const [inline], [final], [override], [virtual]
```

Get the Texture object.

Returns

std::string

Implements [Arcade::IObject](#).

```
25 {return _texture;;}
```

5.3.2.3 getPosition()

```
std::pair< ssize_t, ssize_t > Arcade::AObject::getPosition (
    void ) const [inline], [final], [override], [virtual]
```

Get the Position object.

Returns

std::pair<ssize_t, ssize_t>

Implements [Arcade::IObject](#).

```
22 {return _position;;}
```

5.3.2.4 getShape()

```
Arcade::Shapes Arcade::AObject::getShape (
    void ) const [inline], [final], [override], [virtual]
```

Get the Shape object.

Returns

Arcade::Shapes

Implements [Arcade::IObject](#).

```
21 {return _shape;};
```

5.3.2.5 getSize()

```
std::pair< ssize_t, ssize_t > Arcade::AObject::getSize (
    void ) const [inline], [final], [override], [virtual]
```

Get the Size object.

Returns

std::pair<ssize_t, ssize_t>

Implements [Arcade::IObject](#).

```
23 {return _size;};
```

5.3.2.6 setColor()

```
void Arcade::AObject::setColor (
    Arcade::Colors ) [inline], [final], [override], [virtual]
```

Get the Color object.

Returns

Arcade::Colors

Implements [Arcade::IObject](#).

```
30 {_color = color;};
```

5.3.2.7 setFilePath()

```
void Arcade::AObject::setFilePath (
    std::string ) [inline], [final], [override], [virtual]
```

Set the Texture object.

Parameters

| | |
|----------------|--|
| <i>texture</i> | |
|----------------|--|

Implements [Arcade::IObject](#).

```
31 { _texture = texture;};
```

5.3.2.8 setPosition()

```
void Arcade::AObject::setPosition (
    std::pair< ssize_t, ssize_t > ) [inline], [final], [override], [virtual]
```

Set the Position object.

Parameters

| | |
|-----------------|--|
| <i>position</i> | |
|-----------------|--|

Implements [Arcade::IObject](#).

```
28 { _position = position;};
```

5.3.2.9 setShape()

```
void Arcade::AObject::setShape (
    Arcade::Shapes ) [inline], [final], [override], [virtual]
```

Set the Shape object.

Parameters

| | |
|--------------|--|
| <i>shape</i> | |
|--------------|--|

Implements [Arcade::IObject](#).

```
27 { _shape = shape;};
```

5.3.2.10 setSize()

```
void Arcade::AObject::setSize (
    std::pair< ssize_t, ssize_t > ) [inline], [final], [override], [virtual]
```

Set the Size object.

Parameters

| | |
|-------------|--|
| <i>size</i> | |
|-------------|--|

Implements [Arcade::IObject](#).

```
29 { _size = size; }
```

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

- core/src/Object/AObjects.hpp

5.4 Arcade::Core Class Reference

Public Member Functions

- [Core](#) (std::string libFilePath)
- void [loop](#) ()
- float [getDeltaTime](#) () const

5.4.1 Constructor & Destructor Documentation

5.4.1.1 Core()

```
Arcade::Core::Core (
    std::string libFilePath )
11 {
12     try {
13         storeLibsPath();
14         _lib.second = _lib.first.loadGraphicalLib(libFilePath);
15     } catch (const LoaderException &e) {
16         std::cerr << e.what() << std::endl;
17         exit(84);
18     }
19     _startTime = std::chrono::high_resolution_clock::now();
20     _currentScene = Arcade::Scenes::MAIN_MENU;
21     for (size_t i = 0; i < _libsPath.size(); i++) {
22         if (_libsPath[i] == libFilePath)
23             _currentLib = i;
24     }
25     _currentGame = 0;
26     if (_libsPath.size() == 0 || _gamesPath.size() == 0) {
27         std::cerr << "No library found" << std::endl;
28         exit(84);
29     }
30     _lib.second.get()->createWindow();
31     initMenu();
32 }
```

5.4.1.2 ~Core()

```
Arcade::Core::~Core ( )
35 {
36     _lib.first.closeLib();
37 }
```

5.4.2 Member Function Documentation

5.4.2.1 getDeltaTime()

```
float Arcade::Core::getDeltaTime ( ) const [inline]
40 { return _deltaTime; }
```

5.4.2.2 loop()

```
void Arcade::Core::loop ( )
164 {
165     while ( _currentScene != Arcade::Scenes::LEAVE ) {
166         _lib.second.get ()->updateEvent ();
167         updateDeltaTime ();
168         wait (0.01);
169         _lib.second->clearWindow ();
170         runScene ( _currentScene );
171         _lib.second->renderWindow ();
172     }
173 }
```

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

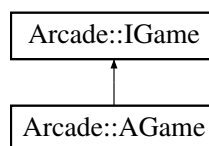
- core/Core.hpp
- core/Core.cpp

5.5 Arcade::!Game Class Reference

Interface for one game library.

```
#include <!Game.hpp>
```

Inheritance diagram for Arcade::!Game:



Public Member Functions

- **IGame** (void)=default
Construct a new [IGame](#) object.
- virtual **~IGame** ()=default
Destroy the [IGame](#) object.
- virtual void **load** (void)=0
Load the game.
- virtual void **update** ([Arcade::ILib](#) &lib, float seconds)=0
Update the game.
- virtual void **render** ([Arcade::ILib](#) &lib)=0
Render the game.
- virtual void **reset** (void)=0
Reset the game.
- virtual void **unload** (void)=0
Unload the game.
- virtual [Arcade::gameState](#) **getState** (void) const =0
Get the game state.
- virtual [ssize_t](#) **getScore** (void) const =0
Get the game score.
- virtual [ssize_t](#) **getHighScore** (void) const =0
Get the game high score.
- virtual [std::string](#) **getGameName** (void) const =0
Get the game name.

5.5.1 Detailed Description

Interface for one game library.

5.5.2 Member Function Documentation

5.5.2.1 getGameName()

```
virtual std::string Arcade::IGame::getGameName (  
    void ) const [pure virtual]
```

Get the game name.

This function will return the name of the game.

Returns

The name of the game.

Implemented in [Arcade::AGame](#).

5.5.2.2 getHighScore()

```
virtual ssize_t Arcade::IGame::getHighScore (
    void ) const [pure virtual]
```

Get the game high score.

This function will return the current high score of the game.

Returns

The current high score of the game.

Implemented in [Arcade::AGame](#).

5.5.2.3 getScore()

```
virtual ssize_t Arcade::IGame::getScore (
    void ) const [pure virtual]
```

Get the game score.

This function will return the current score of the game.

Returns

The current score of the game.

Implemented in [Arcade::AGame](#).

5.5.2.4 getState()

```
virtual Arcade::gameState Arcade::IGame::getState (
    void ) const [pure virtual]
```

Get the game state.

This function will return the current state of the game.

Returns

The current state of the game.

Implemented in [Arcade::AGame](#).

5.5.2.5 load()

```
virtual void Arcade::IGame::load (  
    void ) [pure virtual]
```

Load the game.

This function can load all the assets needed for the game or entities that will be used in the game.

5.5.2.6 render()

```
virtual void Arcade::IGame::render (  
    Arcade::ILib & lib ) [pure virtual]
```

Render the game.

This function will be called every frame and will render entities with graphical library.

Parameters

| | |
|------------|---|
| <i>lib</i> | graphical library, used to render entities. |
|------------|---|

5.5.2.7 reset()

```
virtual void Arcade::IGame::reset (  
    void ) [pure virtual]
```

Reset the game.

This function will be called when the game is reset. It should reset all the entities to their initial state.

5.5.2.8 unload()

```
virtual void Arcade::IGame::unload (  
    void ) [pure virtual]
```

Unload the game.

This function will be called when the game is unloaded. It should unload all the assets and entities.

5.5.2.9 update()

```
virtual void Arcade::IGame::update (  
    Arcade::ILib & lib,  
    float seconds ) [pure virtual]
```

Update the game.

This function will be called every frame and will update the game.

Parameters

| | |
|----------------|--|
| <i>lib</i> | graphical library, used to get inputs. |
| <i>seconds</i> | time elapsed since the last frame. |

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

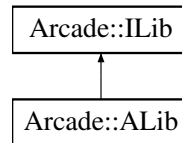
- core/src/Game/IGame.hpp

5.6 Arcade::ILib Class Reference

Interface for the graphical library.

```
#include <ILib.hpp>
```

Inheritance diagram for Arcade::ILib:



Public Member Functions

- **ILib** (void)=default
Construct a new [ILib](#) object.
- **~ILib** ()=default
Destroy the [ILib](#) object.
- virtual bool **isKeyPressed** (Arcade::Inputs input)=0
Check if a key is pressed.
- virtual bool **isKeyReleased** (Arcade::Inputs input)=0
Check if a key is released.
- virtual bool **isWindowClosed** (void)=0
Check if the window is closed.
- virtual void **updateEvent** (void)=0
Update all entities in the window.
- virtual void **createWindow** (void)=0
Create the window and open it.
- virtual void **closeWindow** (void)=0
Close the window.
- virtual void **clearWindow** (void)=0
Clear all entities in the window.
- virtual void **renderWindow** (void)=0
Display all entities in the window.
- virtual void **drawObjects** (std::shared_ptr< [Arcade::IObject](#) > object)=0
Draw an [IObject](#) in the window.
- virtual void **drawShapes** (Arcade::Shapes shape, Arcade::Colors color, std::pair< ssize_t, ssize_t > pos, std::pair< ssize_t, ssize_t > size)=0
Draw a shape in the window with a color, a position and a size.
- virtual void **drawText** (std::shared_ptr< [Arcade::Text](#) > text)=0
Draw a text in the window.
- virtual void **drawText** (std::string str, Arcade::Colors color, ssize_t size, std::pair< ssize_t, ssize_t > pos)=0
Draw a text in the window.
- virtual void **setScale** (std::pair< ssize_t, ssize_t > scale)=0
Set the Scale of the window.
- virtual void **setScale** (ssize_t scale)=0
Set the Scale of the window.
- virtual void **setSize** (std::pair< ssize_t, ssize_t > size)=0
Set the Size of the window.
- virtual std::pair< ssize_t, ssize_t > **getScale** (void) const =0
Get the Scale of the window.
- virtual std::pair< ssize_t, ssize_t > **getSize** (void) const =0
Get the Size of the window.

5.6.1 Detailed Description

Interface for the graphical library.

5.6.2 Member Function Documentation

5.6.2.1 drawObjets()

```
virtual void Arcade::ILib::drawObjets (
    std::shared_ptr< Arcade::IObject > object ) [pure virtual]
```

Draw an [IObject](#) in the window.

Parameters

| | |
|---------------|--------------------|
| <i>object</i> | The object to draw |
|---------------|--------------------|

5.6.2.2 drawShapes()

```
virtual void Arcade::ILib::drawShapes (
    Arcade::Shapes shape,
    Arcade::Colors color,
    std::pair< ssize_t, ssize_t > pos,
    std::pair< ssize_t, ssize_t > size ) [pure virtual]
```

Draw a shape in the window with a color, a position and a size.

Parameters

| | |
|--------------|---------------------------|
| <i>shape</i> | The shape to draw |
| <i>color</i> | The color of the shape |
| <i>pos</i> | The position of the shape |
| <i>size</i> | The size of the shape |

5.6.2.3 drawText() [1/2]

```
virtual void Arcade::ILib::drawText (
    std::shared_ptr< Arcade::Text > text ) [pure virtual]
```

Draw a text in the window.

Parameters

| | |
|-------------|------------------|
| <i>text</i> | The text to draw |
|-------------|------------------|

5.6.2.4 drawText() [2/2]

```
virtual void Arcade::ILib::drawText (
    std::string str,
    Arcade::Colors color,
    ssize_t size,
    std::pair< ssize_t, ssize_t > pos ) [pure virtual]
```

Draw a text in the window.

Parameters

| | |
|--------------|--------------------------|
| <i>str</i> | The text to draw |
| <i>color</i> | The color of the text |
| <i>size</i> | The size of the text |
| <i>pos</i> | The position of the text |

5.6.2.5 getScale()

```
virtual std::pair< ssize_t, ssize_t > Arcade::ILib::getScale (
    void ) const [pure virtual]
```

Get the Scale of the window.

Returns

`std::pair<ssize_t, ssize_t>`

Implemented in [Arcade::ALib](#).

5.6.2.6 getSize()

```
virtual std::pair< ssize_t, ssize_t > Arcade::ILib::getSize (
    void ) const [pure virtual]
```

Get the Size of the window.

Returns

`std::pair<ssize_t, ssize_t>`

Implemented in [Arcade::ALib](#).

5.6.2.7 isKeyPressed()

```
virtual bool Arcade::ILib::isKeyPressed (
    Arcade::Inputs input ) [pure virtual]
```

Check if a key is pressed.

Parameters

| | |
|--------------|------------------|
| <i>input</i> | The key to check |
|--------------|------------------|

Returns

true if the key is pressed, false otherwise

Implemented in [Arcade::ALib](#).

5.6.2.8 isKeyReleased()

```
virtual bool Arcade::ILib::isKeyReleased (
    Arcade::Inputs input ) [pure virtual]
```

Check if a key is released.

Parameters

| | |
|--------------|------------------|
| <i>input</i> | The key to check |
|--------------|------------------|

Returns

true if the key is released, false otherwise

Implemented in [Arcade::ALib](#).

5.6.2.9 isWindowClosed()

```
virtual bool Arcade::ILib::isWindowClosed (
    void ) [pure virtual]
```

Check if the window is closed.

Returns

true if the window is closed, false otherwise

5.6.2.10 setScale() [1/2]

```
virtual void Arcade::ILib::setScale (
    ssize_t scale ) [pure virtual]
```

Set the Scale of the window.

Parameters

| | |
|--------------|------------------|
| <i>scale</i> | The scale to set |
|--------------|------------------|

Implemented in [Arcade::ALib](#).

5.6.2.11 setScale() [2/2]

```
virtual void Arcade::ILib::setScale (
    std::pair< ssize_t, ssize_t > scale ) [pure virtual]
```

Set the Scale of the window.

Parameters

| | |
|--------------|------------------|
| <i>scale</i> | The scale to set |
|--------------|------------------|

Implemented in [Arcade::ALib](#).

5.6.2.12 setSize()

```
virtual void Arcade::ILib::setSize (
    std::pair< ssize_t, ssize_t > size ) [pure virtual]
```

Set the Size of the window.

Parameters

| | |
|-------------|-----------------|
| <i>size</i> | The size to set |
|-------------|-----------------|

Implemented in [Arcade::ALib](#).

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

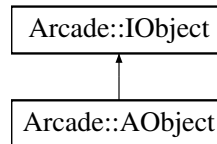
- core/src/Lib/ILib.hpp

5.7 Arcade::IObject Class Reference

Interface for the object.

```
#include <IObject.hpp>
```

Inheritance diagram for Arcade::IObject:



Public Member Functions

- [IObject](#) (void)=default
Construct a new [IObject](#) object.
- virtual [~IObject](#) ()=default
Destroy the [IObject](#) object.
- virtual Arcade::Shapes [getShape](#) (void) const =0
Get the Shape object.
- virtual void [setShape](#) (Arcade::Shapes)=0
Set the Shape object.
- virtual std::pair< ssize_t, ssize_t > [getPosition](#) (void) const =0
Get the Position object.
- virtual void [setPosition](#) (std::pair< ssize_t, ssize_t >)=0
Set the Position object.
- virtual std::pair< ssize_t, ssize_t > [getSize](#) (void) const =0
Get the Size object.
- virtual void [setSize](#) (std::pair< ssize_t, ssize_t >)=0
Set the Size object.
- virtual void [setColor](#) (Arcade::Colors)=0
Get the Color object.
- virtual Arcade::Colors [getColor](#) (void) const =0
Set the Color object.
- virtual std::string [getFilePath](#) (void) const =0
Get the Texture object.
- virtual void [setFilePath](#) (std::string)=0
Set the Texture object.

5.7.1 Detailed Description

Interface for the object.

The methods are pure virtual and must be overridden in the inherited classes

5.7.2 Constructor & Destructor Documentation

5.7.2.1 IObject()

```
Arcade::IObject::IObject (
    void ) [default]
```

Construct a new [IObject](#) object.

It is used to draw something with graphical library

5.7.3 Member Function Documentation

5.7.3.1 getColor()

```
virtual Arcade::Colors Arcade::IObject::getColor (
    void ) const [pure virtual]
```

Set the Color object.

Parameters

| | |
|--------------|--|
| <i>color</i> | |
|--------------|--|

Implemented in [Arcade::AObject](#).

5.7.3.2 getFilePath()

```
virtual std::string Arcade::IObject::getFilePath (
    void ) const [pure virtual]
```

Get the Texture object.

Returns

std::string

Implemented in [Arcade::AObject](#).

5.7.3.3 getPosition()

```
virtual std::pair< ssize_t, ssize_t > Arcade::IObject::getPosition (
    void ) const [pure virtual]
```

Get the Position object.

Returns

std::pair<ssize_t, ssize_t>

Implemented in [Arcade::AObject](#).

5.7.3.4 getShape()

```
virtual Arcade::Shapes Arcade::IObject::getShape (
    void ) const [pure virtual]
```

Get the Shape object.

Returns

Arcade::Shapes

Implemented in [Arcade::AObject](#).

5.7.3.5 getSize()

```
virtual std::pair< ssize_t, ssize_t > Arcade::IObject::getSize (
    void ) const [pure virtual]
```

Get the Size object.

Returns

std::pair<ssize_t, ssize_t>

Implemented in [Arcade::AObject](#).

5.7.3.6 setColor()

```
virtual void Arcade::IObject::setColor (
    Arcade::Colors ) [pure virtual]
```

Get the Color object.

Returns

Arcade::Colors

Implemented in [Arcade::AObject](#).

5.7.3.7 setFilePath()

```
virtual void Arcade::IObject::setFilePath (
    std::string ) [pure virtual]
```

Set the Texture object.

Parameters

| | |
|----------------|--|
| <i>texture</i> | |
|----------------|--|

Implemented in [Arcade::AObject](#).

5.7.3.8 setPosition()

```
virtual void Arcade::IObject::setPosition (
    std::pair< ssize_t, ssize_t > ) [pure virtual]
```

Set the Position object.

Parameters

| | |
|-----------------|--|
| <i>position</i> | |
|-----------------|--|

Implemented in [Arcade::AObject](#).

5.7.3.9 setShape()

```
virtual void Arcade::IObject::setShape (
    Arcade::Shapes ) [pure virtual]
```

Set the Shape object.

Parameters

| | |
|--------------|--|
| <i>shape</i> | |
|--------------|--|

Implemented in [Arcade::AObject](#).

5.7.3.10 setSize()

```
virtual void Arcade::IObject::setSize (
    std::pair< ssize_t, ssize_t > ) [pure virtual]
```

Set the Size object.

Parameters

| | |
|-------------|--|
| <i>size</i> | |
|-------------|--|

Implemented in [Arcade::AObject](#).

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

- `core/src/Object/Object.hpp`

5.8 Arcade::LibLoader< LibInterface > Class Template Reference

Public Member Functions

- `std::shared_ptr< LibInterface > loadGraphicalLib (const std::string &libPath)`
Loads a library.
- `std::shared_ptr< LibInterface > loadGameLib (const std::string &libPath)`
- `void closeLib ()`
- `bool isLibOpen () const`

5.8.1 Constructor & Destructor Documentation

5.8.1.1 LibLoader()

```
template<typename LibInterface >
Arcade::LibLoader< LibInterface >::LibLoader (
    void ) [inline]
24 {};
```

5.8.1.2 ~LibLoader()

```
template<typename LibInterface >
Arcade::LibLoader< LibInterface >::~~LibLoader ( ) [inline]
25 {};
```

5.8.2 Member Function Documentation

5.8.2.1 closeLib()

```
template<typename LibInterface >
void Arcade::LibLoader< LibInterface >::closeLib ( ) [inline]
58     {
59         if (dlclose(_handle) != 0)
60             throw LoaderException(dlerror());
61     }
```

5.8.2.2 isLibOpen()

```
template<typename LibInterface >
bool Arcade::LibLoader< LibInterface >::isLibOpen ( ) const [inline]
64     {
65         return _handle != nullptr;
66     }
```

5.8.2.3 loadGameLib()

```
template<typename LibInterface >
std::shared_ptr< LibInterface > Arcade::LibLoader< LibInterface >::loadGameLib (
    const std::string & libPath ) [inline]
46     {
47         _handle = dlopen(libPath.c_str(), RTLD_LAZY);
48         LibInterface *(*builder)() = nullptr;
49         if (!_handle)
50             throw LoaderException(dlerror());
51         builder = reinterpret_cast<LibInterface *(*)()>(dlsym(_handle, "GameEntryPoint"));
52         if (!builder)
53             throw LoaderException(dlerror());
54         return std::shared_ptr<LibInterface>(builder());
55     }
```

5.8.2.4 loadGraphicalLib()

```
template<typename LibInterface >
std::shared_ptr< LibInterface > Arcade::LibLoader< LibInterface >::loadGraphicalLib (
    const std::string & libPath ) [inline]
```

Loads a library.

Parameters

| | |
|----------------|---------------------------------|
| <i>libPath</i> | The name of the library to load |
|----------------|---------------------------------|

Returns

A shared pointer to the library

The library must be in the ./lib folder

```
34     {
35         _handle = dlopen(libPath.c_str(), RTLD_LAZY);
36         LibInterface *(*builder)() = nullptr;
37         if (!_handle)
38             throw LoaderException(dlerror());
39         builder = reinterpret_cast<LibInterface *(*)()>(dlsym(_handle, "DisplayEntryPoint"));
40         if (!builder)
41             throw LoaderException(dlerror());
42         return std::shared_ptr<LibInterface>(builder());
43     }
```

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

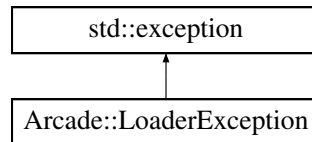
- core/LibLoader.hpp

5.9 Arcade::LoaderException Class Reference

The class [LoaderException](#) is the exception thrown when the library cannot be loaded.

```
#include <ErrorManagement.hpp>
```

Inheritance diagram for Arcade::LoaderException:



Public Member Functions

- [LoaderException](#) (const std::string &message)
- const char * [what](#) () const throw ()

5.9.1 Detailed Description

The class [LoaderException](#) is the exception thrown when the library cannot be loaded.

Parameters

| | |
|----------------|------------------------|
| <i>message</i> | The message to display |
|----------------|------------------------|

5.9.2 Constructor & Destructor Documentation

5.9.2.1 LoaderException()

```
Arcade::LoaderException::LoaderException (
    const std::string & message ) [inline]
20 : _message(message) {}
```

5.9.2.2 ~LoaderException()

```
Arcade::LoaderException::~~LoaderException ( ) throw ( ) [inline]
21 {}
```

5.9.3 Member Function Documentation

5.9.3.1 what()

```
const char * Arcade::LoaderException::what ( ) const throw ( )    [inline]
22 { return _message.c_str(); }
```

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

- core/ErrorMessageManagement.hpp

5.10 Arcade::Text Class Reference

The class [Text](#) is the generic way to handle text.

```
#include <Text.hpp>
```

Public Member Functions

- [Text](#) (void)
Construct a new [Text](#) object.
- [Text](#) (std::string text, std::pair< ssize_t, ssize_t > pos, Arcade::Colors color)
Construct a new [Text](#) object.
- [~Text](#) ()=default
Destroy the [Text](#) object.
- std::string [getText](#) (void) const
Get the [Text](#) object.
- void [setText](#) (std::string)
Set the [Text](#) object.
- std::pair< ssize_t, ssize_t > [getPosition](#) (void) const
Get the [Position](#) object.
- void [setPosition](#) (std::pair< ssize_t, ssize_t >)
Set the [Position](#) object.
- void [setColor](#) (Arcade::Colors)
Get the [Color](#) object.
- Arcade::Colors [getColor](#) (void) const
Set the [Color](#) object.

5.10.1 Detailed Description

The class [Text](#) is the generic way to handle text.

5.10.2 Constructor & Destructor Documentation

5.10.2.1 Text() [1/2]

```
Arcade::Text::Text (
    void )
```

Construct a new [Text](#) object.

The default constructor of the class [Text](#)

Must be implemented in the inherited classes

5.10.2.2 Text() [2/2]

```
Arcade::Text::Text (
    std::string text,
    std::pair< ssize_t, ssize_t > pos,
    Arcade::Colors color )
```

Construct a new [Text](#) object.

The constructor of the class [Text](#)

Must be implemented in the inherited classes

Parameters

| | |
|--------------|--------------------------|
| <i>text</i> | The text to display |
| <i>pos</i> | The position of the text |
| <i>color</i> | The color of the text |

5.10.3 Member Function Documentation

5.10.3.1 getColor()

```
Arcade::Colors Arcade::Text::getColor (
    void ) const [inline]
```

Set the Color object.

Parameters

| | |
|--------------|--|
| <i>color</i> | |
|--------------|--|


```
77 { return _color; };
```

5.10.3.2 getPosition()

```
std::pair< ssize_t, ssize_t > Arcade::Text::getPosition (
    void ) const [inline]
```

Get the Position object.

Returns

std::pair<ssize_t, ssize_t>

```
59 { return _position; };
```

5.10.3.3 getText()

```
std::string Arcade::Text::getText (
    void ) const [inline]
```

Get the [Text](#) object.

Returns

std::string

```
47 { return _text;};
```

5.10.3.4 setColor()

```
void Arcade::Text::setColor (
    Arcade::Colors ) [inline]
```

Get the Color object.

Returns

Arcade::Colors

```
71 { _color = _color; };
```

5.10.3.5 setPosition()

```
void Arcade::Text::setPosition (
    std::pair< ssize_t, ssize_t > ) [inline]
```

Set the Position object.

Parameters

| | |
|-----------------|--|
| <i>position</i> | |
|-----------------|--|

```
65 { _position = _position; };
```

5.10.3.6 setText()

```
void Arcade::Text::setText (
    std::string ) [inline]
```

Set the [Text](#) object.

Parameters

| | |
|-------------|--|
| <i>text</i> | |
|-------------|--|

```
53 { _text = _text; };
```

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

- core/src/Text/Text.hpp

Chapter 6

File Documentation

6.1 Core.hpp

```
1  /*
2  ** EPITECH PROJECT, 2023
3  ** B-OOP-400-NAN-4-1-arcade-architecture
4  ** File description:
5  ** Arcade
6  */
7
8  #pragma once
9  #include <iostream>
10 #include <string>
11 #include <memory>
12 #include <map>
13 #include <vector>
14 #include <filesystem>
15 #include <algorithm>
16 #include <time.h>
17 #include <chrono>
18
19 #include "IGame.hpp"
20 #include "Lib/ILib.hpp"
21 #include "LibLoader.hpp"
22 #include "AObjects.hpp"
23
24 namespace Arcade {
25     /*Those are just example */
26
27     enum Scenes {
28         MAIN_MENU,
29         IN_GAME,
30         LEAVE
31     };
32
33     class Core {
34     public:
35         Core(std::string libFilePath);
36         ~Core();
37
38         void loop();
39
40         float getDeltaTime() const { return _deltaTime; };
41
42     private:
43         void runScene(Arcade::Scenes scene = Arcade::MAIN_MENU);
44         bool loadGame(const std::string &GameName);
45         bool loadLib(const std::string &LibName);
46         void updateDeltaTime(void);
47
48         void updateMainMenu(Arcade::ILib &lib);
49         void renderMainMenu(Arcade::ILib &lib);
50         void initMenu();
51         void wait(double time);
52
53         void globalInputs(Arcade::ILib &lib);
54
55         std::vector<std::string> getLibsFromDirectory();
56         void storeLibsPath();
57
58         std::pair<Arcade::LibLoader<Arcade::ILib>, std::shared_ptr<Arcade::ILib> _lib;
```

```

59         std::pair<Arcade::LibLoader<Arcade::IGame>, std::shared_ptr<Arcade::IGame> _game;
60
61         std::vector<std::string> _gamesPath;
62         std::vector<std::string> _libsPath;
63
64         std::size_t _currentGame;
65         std::size_t _currentLib;
66         Arcade::Scenes _currentScene;
67
68         double _deltaTime;
69         std::chrono::V2::system_clock::time_point _startTime;
70
71         std::vector<std::shared_ptr<Arcade::IObject> _menuObjects;
72     };
73 };

```

6.2 ErrorManagement.hpp

```

1  /*
2  ** EPITECH PROJECT, 2023
3  ** B-OOP-400-NAN-4-1-arcade-architecture
4  ** File description:
5  ** ErrorManagement
6  */
7
8  #pragma once
9  #include <exception>
10 #include <string>
11
12 namespace Arcade
13 {
14     class LoaderException : public std::exception {
15     public:
16         LoaderException(const std::string &message) : _message(message) {}
17         ~LoaderException() throw() {}
18         const char *what() const throw() { return _message.c_str(); }
19     private:
20         std::string _message;
21     };
22 } // namespace Arcade

```

6.3 LibLoader.hpp

```

1  /*
2  ** EPITECH PROJECT, 2023
3  ** B-OOP-400-NAN-4-1-arcade-architecture
4  ** File description:
5  ** LibLoader
6  */
7
8  #pragma once
9  #include <string>
10 #include <memory.h>
11 #include <dlfcn.h>
12 #include "ErrorManagement.hpp"
13
14 namespace Arcade {
15     template <typename LibInterface>
16     class LibLoader {
17     public:
18         LibLoader(void) {};
19         ~LibLoader() {};
20
21         std::shared_ptr<LibInterface> loadGraphicalLib(const std::string &libPath)
22         {
23             _handle = dlopen(libPath.c_str(), RTLD_LAZY);
24             LibInterface *(*builder)() = nullptr;
25             if (!_handle)
26                 throw LoaderException(dlerror());
27             builder = reinterpret_cast<LibInterface *(*)()>(dlsym(_handle, "DisplayEntryPoint"));
28             if (!builder)
29                 throw LoaderException(dlerror());
30             return std::shared_ptr<LibInterface>(builder());
31         }
32
33         std::shared_ptr<LibInterface> loadGameLib(const std::string &libPath)
34         {
35             _handle = dlopen(libPath.c_str(), RTLD_LAZY);
36             LibInterface *(*builder)() = nullptr;

```

```

49         if (!_handle)
50             throw LoaderException(dlerror());
51         builder = reinterpret_cast<LibInterface *(&)()>(dlsym(_handle, "GameEntryPoint"));
52         if (!builder)
53             throw LoaderException(dlerror());
54         return std::shared_ptr<LibInterface>(builder());
55     }
56
57     void closeLib()
58     {
59         if (dlclose(_handle) != 0)
60             throw LoaderException(dlerror());
61     }
62
63     bool isLibOpen() const
64     {
65         return _handle != nullptr;
66     }
67
68     private:
69         void *_handle;
70 };
71 };

```

6.4 AGame.hpp

```

1  /*
2  ** EPITECH PROJECT, 2023
3  ** B-OOP-400-NAN-4-1-arcade-architecture
4  ** File description:
5  ** AGame
6  */
7
8  #pragma once
9  #include "IGame.hpp"
10
11 namespace Arcade
12 {
13     class AGame : virtual public Arcade::IGame {
14     public:
15         AGame(void) = default;
16         ~AGame() = default;
17
18         Arcade::gameState getState(void) const final override {return _state;};
19         ssize_t getScore(void) const final override {return _score;};
20         ssize_t getHighScore(void) const final override {return _highScore;};
21         std::string getGameName(void) const final override {return _name;};
22
23     protected:
24         std::string _name;
25         Arcade::gameState _state;
26         ssize_t _score;
27         ssize_t _highScore;
28     };
29 }

```

6.5 IGame.hpp

```

1  /*
2  ** EPITECH PROJECT, 2023
3  ** arcade-archi
4  ** File description:
5  ** IGame
6  */
7
8  #pragma once
9
10 #include <iostream>
11 #include <string>
12 #include <memory>
13 #include "Lib/ILib.hpp"
14
15 namespace Arcade {
16     enum gameState {
17         MENU,
18         GAME,
19         PAUSE,
20         END
21     };

```

```

22
23
24
25
26     class IGame {
27     public:
28
29         IGame(void) = default;
30
31         virtual ~IGame() = default;
32
33         virtual void load(void) = 0;
34
35         virtual void update(Arcade::ILib &lib, float seconds) = 0;
36
37         virtual void render(Arcade::ILib &lib) = 0;
38
39         virtual void reset(void) = 0;
40
41         virtual void unload(void) = 0;
42
43         virtual Arcade::gameState getState(void) const = 0;
44
45         virtual ssize_t getScore(void) const = 0;
46
47         virtual ssize_t getHighScore(void) const = 0;
48
49         virtual std::string getGameName(void) const = 0;
50     };
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

6.6 ALib.hpp

```

1  /*
2  ** EPITECH PROJECT, 2023
3  ** B-OOP-400-NAN-4-1-arcade-architecture
4  ** File description:
5  ** ALib
6  */
7
8  #pragma once
9  #include "ILib.hpp"
10 #include "Types.hpp"
11 #include <unordered_map>
12
13 namespace Arcade {
14     class ALib : virtual public Arcade::ILib {
15     public:
16
17         ALib(void) = default;
18         ~ALib() = default;
19
20         void setScale(std::pair<ssize_t, ssize_t> scale) final override { _scale = scale; }
21         void setScale(ssize_t scale) final override { _scale = std::pair<ssize_t, ssize_t>(scale,
22         scale); }
23
24         void setSize(std::pair<ssize_t, ssize_t> size) final override { _size = size; }
25         std::pair<ssize_t, ssize_t> getScale(void) const final override { return _scale; }
26         std::pair<ssize_t, ssize_t> getSize(void) const final override { return _size; }
27
28         bool isKeyPressed(Arcade::Inputs) override { return false; }
29         bool isKeyReleased(Arcade::Inputs) override { return false; }
30
31     private:
32         std::pair<ssize_t, ssize_t> _scale;
33         std::pair<ssize_t, ssize_t> _size;
34
35     protected:
36         std::unordered_map<Arcade::Inputs, bool> _keys;
37         std::string _name;
38     };
39 }
40
41
42
43

```

6.7 ILib.hpp

```

1  /*
2  ** EPITECH PROJECT, 2023
3  ** B-OOP-400-NAN-4-1-arcade-architecture
4  ** File description:
5  ** ILib
6  */
7
8  #pragma once
9  #include "Types.hpp"
10 #include <memory>

```

```

11 #include "Text.hpp"
12 #include "IOObject.hpp"
13
14 namespace Arcade {
15     class ILib {
16     public:
17         ILib(void) = default;
18
19         ~ILib() = default;
20
21         virtual bool isKeyPressed(Arcade::Inputs input) = 0;
22
23         virtual bool isKeyReleased(Arcade::Inputs input) = 0;
24
25         virtual bool isWindowClosed(void) = 0;
26
27         virtual void updateEvent(void) = 0;
28
29         virtual void createWindow(void) = 0;
30
31         virtual void closeWindow(void) = 0;
32
33         virtual void clearWindow(void) = 0;
34
35         virtual void renderWindow(void) = 0;
36
37         virtual void drawObjets(std::shared_ptr<Arcade::IOObject> object) = 0;
38         virtual void drawShapes(Arcade::Shapes shape, Arcade::Colors color, std::pair<ssize_t,
39             ssize_t> pos, std::pair<ssize_t, ssize_t> size) = 0;
40
41         virtual void drawText(std::shared_ptr<Arcade::Text> text) = 0;
42         virtual void drawText(std::string str, Arcade::Colors color, ssize_t size,
43             std::pair<ssize_t, ssize_t> pos) = 0;
44
45         virtual void setScale(std::pair<ssize_t, ssize_t> scale) = 0;
46
47         virtual void setScale(ssize_t scale) = 0;
48
49         virtual void setSize(std::pair<ssize_t, ssize_t> size) = 0;
50
51         virtual std::pair<ssize_t, ssize_t> getScale(void) const = 0;
52
53         virtual std::pair<ssize_t, ssize_t> getSize(void) const = 0;
54     };
55 }

```

6.8 Types.hpp

```

1  /*
2  ** EPITECH PROJECT, 2023
3  ** B-OOP-400-NAN-4-1-arcade-architecture
4  ** File description:
5  ** Types
6  */
7
8 #pragma once
9
10 namespace Arcade {
11     enum Shapes {
12         SQUARE,
13         CIRCLE,
14         TRIANGLE,
15         NO_SHAPE
16     };
17
18     enum Inputs {
19         IKEY_UP,
20         IKEY_DOWN,
21         IKEY_LEFT,
22         IKEY_RIGHT,
23         IKEY_SPACE,
24         IKEY_ENTER,
25         IKEY_BACKSPACE,
26         IKEY_TAB,
27         IKEY_SHIFT,
28         IKEY_CTRL,
29         IKEY_ALT,
30
31         /* Globals inputs */
32         IKEY_B, //Graphical library
33         IKEY_D, //Game library
34         IKEY_S, //QUIT
35         IKEY_M, //MENU

```

```

37
38     /*Those key are for arcade control Exit/Menu/... */
39     IKEY_Q,
40     IKEY_ESC,
41     IKEY_A,
42     IKEY_C,
43     IKEY_E,
44     IKEY_F,
45     IKEY_G,
46     IKEY_H,
47
48     IKEY_I,
49     IKEY_J,
50     IKEY_K,
51     IKEY_L,
52     IKEY_N,
53     IKEY_O,
54     IKEY_P,
55     IKEY_R,
56     IKEY_T,
57     IKEY_U,
58     IKEY_V,
59     IKEY_W,
60     IKEY_X,
61     IKEY_Y,
62     IKEY_Z,
63     NO_KEY
64 };
65
66 enum Colors {
67     BLACK,
68     WHITE,
69     RED,
70     GREEN,
71     BLUE,
72     YELLOW,
73     MAGENTA,
74     CYAN,
75     TRANSPARENT,
76     NO_COLOR
77 };
78 }

```

6.9 AObjects.hpp

```

1  /*
2  ** EPITECH PROJECT, 2023
3  ** B-OOP-400-NAN-4-1-arcade-architecture
4  ** File description:
5  ** AObjects
6  */
7
8  #pragma once
9  #include "IObject.hpp"
10 #include "Lib/Types.hpp"
11
12 namespace Arcade {
13     class AObject : virtual public Arcade::IObject {
14     public:
15         AObject(void) = default;
16         ~AObject() = default;
17
18         Arcade::Shapes getShape(void) const final override {return _shape;};
19         std::pair<ssize_t, ssize_t> getPosition(void) const final override {return _position;};
20         std::pair<ssize_t, ssize_t> getSize(void) const final override {return _size;};
21         Arcade::Colors getColor(void) const final override {return _color;};
22         std::string getFilePath(void) const final override {return _texture;};
23
24         void setShape(Arcade::Shapes shape) final override {_shape = shape;};
25         void setPosition(std::pair<ssize_t, ssize_t> position) final override {_position =
26             position;};
27         void setSize(std::pair<ssize_t, ssize_t> size) final override {_size = size;};
28         void setColor(Arcade::Colors color) final override {_color = color;};
29         void setFilePath(std::string texture) final override {_texture = texture;};
30
31     protected:
32         Arcade::Shapes _shape;
33         std::pair<ssize_t, ssize_t> _position;
34         std::pair<ssize_t, ssize_t> _size;
35         Arcade::Colors _color;
36         std::string _texture;
37     };
38 }

```


6.10 IObject.hpp

```

1  /*
2  ** EPITECH PROJECT, 2023
3  ** arcade-archi
4  ** File description:
5  ** IObject
6  */
7
8  #pragma once
9  #include <utility>
10 #include "Lib/Types.hpp"
11
12 namespace Arcade {
13
14     class IObject {
15     public:
16         IObject(void) = default;
17
18         virtual ~IObject() = default;
19
20         virtual Arcade::Shapes getShape(void) const = 0;
21
22         virtual void setShape(Arcade::Shapes) = 0;
23
24         virtual std::pair<ssize_t, ssize_t> getPosition(void) const = 0;
25
26         virtual void setPosition(std::pair<ssize_t, ssize_t>) = 0;
27
28         virtual std::pair<ssize_t, ssize_t> getSize(void) const = 0;
29
30         virtual void setSize(std::pair<ssize_t, ssize_t>) = 0;
31
32         virtual void setColor(Arcade::Colors) = 0;
33
34         virtual Arcade::Colors getColor(void) const = 0;
35
36         virtual std::string getFilePath(void) const = 0;
37
38         virtual void setFilePath(std::string) = 0;
39     };
40
41 };

```

6.11 Text.hpp

```

1  /*
2  ** EPITECH PROJECT, 2023
3  ** arcade-archi
4  ** File description:
5  ** IText
6  */
7
8  #pragma once
9  #include <string>
10 #include <utility>
11 #include "Lib/Types.hpp"
12
13 namespace Arcade {
14
15     class Text {
16     public:
17         Text(void);
18
19         Text(std::string text, std::pair<ssize_t, ssize_t> pos, Arcade::Colors color);
20
21         ~Text() = default;
22
23         std::string getText(void) const { return _text; };
24
25         void setText(std::string) { _text = _text; };
26
27         std::pair<ssize_t, ssize_t> getPosition(void) const { return _position; };
28
29         void setPosition(std::pair<ssize_t, ssize_t>) { _position = _position; };
30
31         void setColor(Arcade::Colors) { _color = _color; };
32
33         Arcade::Colors getColor(void) const { return _color; };
34
35     private:
36         std::string _text;
37         std::pair<ssize_t, ssize_t> _position;
38     };
39
40 };

```

```
85         Arcade::Colors _color;  
86     };  
87 };
```