# 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

# Chapter 2

# Hierarchical Index

## 2.1 Class Hierarchy

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

# Chapter 3

# Class Index

## 3.1 Class List

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

# Chapter 4

# File Index

## 4.1 File List

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

# Chapter 5

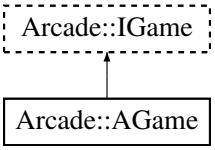
# Class Documentation

## 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:

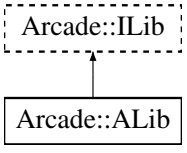
- core/src/Game/AGame.hpp

## 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**

| | |
|---|---|
| *input* | 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**

| | |
|---|---|
| *input* | 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**

| | |
|---|---|
| *scale* | 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**

| | |
|---|---|
| *scale* | 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**

| | |
|---|---|
| *size* | The size to set |

Implements Arcade::ILib.
```
24 { _size = size; }
```

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

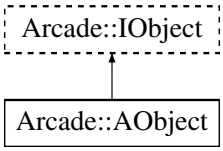
- core/src/Lib/ALib.hpp

## 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**

| | |
|---|---|
| *color* | |

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**

| | |
|---|---|
| *texture* | |

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**

| | |
|---|---|
| *position* | |

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**

| | |
|---|---|
| *shape* | |

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**

| | |
|---|---|
| *size* | |

Implements Arcade::IObject.

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

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

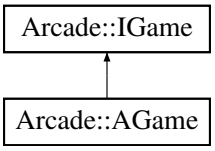
- core/src/Object/AObjects.hpp

## 5.4 Arcade::IGame Class Reference

Interface for one game library.

`#include <IGame.hpp>`

Inheritance diagram for Arcade::IGame:




### 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.*

### 5.4.1 Detailed Description

Interface for one game library.

### 5.4.2 Member Function Documentation

#### 5.4.2.1 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.

#### 5.4.2.2 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.

#### 5.4.2.3 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.

#### 5.4.2.4 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.

#### 5.4.2.5 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.

#### 5.4.2.6 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 grahical library.

**Parameters**

| | |
|---|---|
| *lib* | graphical library, used to render entities. |

#### 5.4.2.7 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.

#### 5.4.2.8 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.

#### 5.4.2.9 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**

| | |
|---|---|
| *lib* | graphical library, used to get inputs. |
| *seconds* | time elapsed since the last frame. |

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

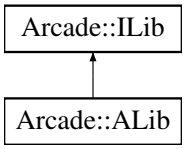
- core/src/Game/IGame.hpp

## 5.5 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 drawObjets (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.*

### 5.5.1 Detailed Description

Interface for the graphical library.

### 5.5.2 Member Function Documentation

#### 5.5.2.1 drawObjets()

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

Draw an IObject in the window.

**Parameters**

| | |
|---|---|
| *object* | The object to draw |

#### 5.5.2.2 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**

| | |
|---|---|
| *shape* | The shape to draw |
| *color* | The color of the shape |
| *pos* | The position of the shape |
| *size* | The size of the shape |

#### 5.5.2.3 drawText() [1/2]

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

Draw a text in the window.

**Parameters**

| | |
|---|---|
| *text* | The text to draw |

#### 5.5.2.4 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**

| | |
|---|---|
| *str* | The text to draw |
| *color* | The color of the text |
| *size* | The size of the text |
| *pos* | The position of the text |

#### 5.5.2.5 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.

#### 5.5.2.6 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.

#### 5.5.2.7 isKeyPressed()

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

Check if a key is pressed.

**Parameters**

| | |
|---|---|
| *input* | The key to check |

**Returns**

true if the key is pressed, false otherwise

Implemented in Arcade::ALib.

#### 5.5.2.8 isKeyReleased()

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

Check if a key is released.

**Parameters**

| | |
|---|---|
| *input* | The key to check |

**Returns**

true if the key is released, false otherwise

Implemented in Arcade::ALib.

#### 5.5.2.9 isWindowClosed()

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

Check if the window is closed.

**Returns**

true if the window is closed, false otherwise

#### 5.5.2.10 setScale() [1/2]

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

Set the Scale of the window.

**Parameters**

| | |
|---|---|
| *scale* | The scale to set |

Implemented in Arcade::ALib.

#### 5.5.2.11 setScale() [2/2]

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

Set the Scale of the window.

**Parameters**

| | |
|---|---|
| *scale* | The scale to set |

Implemented in Arcade::ALib.

#### 5.5.2.12 setSize()

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

Set the Size of the window.

**Parameters**

| | |
|---|---|
| *size* | The size to set |

Implemented in Arcade::ALib.

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

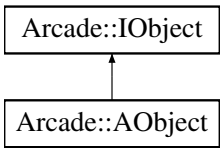
- core/src/Lib/ILib.hpp

## 5.6 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.*

### 5.6.1 Detailed Description

Interface for the object.

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

## 5.6.2 Constructor & Destructor Documentation

### 5.6.2.1 IObject()

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

Construct a new IObject object.

It is used to draw something with graphical library

## 5.6.3 Member Function Documentation

### 5.6.3.1 getColor()

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

Set the Color object.

**Parameters**

| | |
|---|---|
| *color* | |

Implemented in Arcade::AObject.

### 5.6.3.2 getFilePath()

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

Get the Texture object.

**Returns**

std::string

Implemented in Arcade::AObject.

#### 5.6.3.3 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.

#### 5.6.3.4 getShape()

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

Get the Shape object.

**Returns**

Arcade::Shapes

Implemented in Arcade::AObject.

#### 5.6.3.5 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.

#### 5.6.3.6 setColor()

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

Get the Color object.

**Returns**

Arcade::Colors

Implemented in Arcade::AObject.

#### 5.6.3.7 setFilePath()

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

Set the Texture object.

**Parameters**

| | |
|---|---|
| *texture* | |

Implemented in Arcade::AObject.

#### 5.6.3.8 setPosition()

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

Set the Position object.

**Parameters**

| | |
|---|---|
| *position* | |

Implemented in Arcade::AObject.

#### 5.6.3.9 setShape()

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

Set the Shape object.

**Parameters**

| | |
|---|---|
| *shape* | |

Implemented in Arcade::AObject.

#### 5.6.3.10 setSize()

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

Set the Size object.

**Parameters**

| | |
|---|---|
| *size* | |

Implemented in Arcade::AObject.

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

- core/src/Object/IObject.hpp

## 5.7 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.*

### 5.7.1 Detailed Description

The class Text is the generic way to handle text.

### 5.7.2 Constructor & Destructor Documentation

#### 5.7.2.1 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

#### 5.7.2.2 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**

| | |
|---|---|
| *text* | The text to display |
| *pos* | The position of the text |
| *color* | The color of the text |

### 5.7.3 Member Function Documentation

#### 5.7.3.1 getColor()

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

Set the Color object.

**Parameters**

| | |
|---|---|
| *color* | |

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

#### 5.7.3.2 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; };
```

#### 5.7.3.3 getText()

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

Get the Text object.

**Returns**

std::string

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

#### 5.7.3.4 setColor()

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

Get the Color object.

**Returns**

Arcade::Colors

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

#### 5.7.3.5 setPosition()

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

Set the Position object.

**Parameters**

| | |
|---|---|
| *position* | |

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

#### 5.7.3.6 setText()

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

Set the Text object.

**Parameters**

| | |
|---|---|
| *text* | |

```
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 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 {
16     class AGame : virtual public Arcade::IGame {
17         public:
18             AGame(void) = default;
19             ~AGame() = default;
20
21             Arcade::gameState getState(void) const final override {return _state;};
22             ssize_t getScore(void) const final override {return _score;};
23             ssize_t getHighScore(void) const final override {return _highScore;};
24             std::string getGameName(void) const final override {return _name;};
25
26         protected:
28             std::string _name;
30             Arcade::gameState _state;
32             ssize_t _score;
34             ssize_t _highScore;
35     };
36 }
```

## 6.2 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
```

```
26      class IGame {
27          public:
31              IGame(void) = default;
32
36              virtual ~IGame() = default;
37
42              virtual void load(void) = 0;
43
50              virtual void update(Arcade::ILib &lib, float seconds) = 0;
51
57              virtual void render(Arcade::ILib &lib) = 0;
58
63              virtual void reset(void) = 0;
64
69              virtual void unload(void) = 0;
70
76              virtual Arcade::gameState getState(void) const = 0;
77
83              virtual ssize_t getScore(void) const = 0;
84
90              virtual ssize_t getHighScore(void) const = 0;
91
97              virtual std::string getGameName(void) const = 0;
98      };
99  };
```

## 6.3 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 {
17      class ALib : virtual public Arcade::ILib {
18          public:
19              ALib(void) = default;
20              ~ALib() = default;
21
22              void setScale(std::pair<ssize_t, ssize_t> scale) final override { _scale = scale; }
23              void setScale(ssize_t scale) final override { _scale = std::pair<ssize_t, ssize_t>(scale,
        scale); }
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:
33              std::pair<ssize_t, ssize_t> _scale;
35              std::pair<ssize_t, ssize_t> _size;
36
37          protected:
39              std::unordered_map<Arcade::Inputs, bool> _keys;
41              std::string _name;
42      };
43  }
```

## 6.4 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 "IObject.hpp"
13
14 namespace Arcade {
18     class ILib {
19         public:
23             ILib(void) = default;
24
28             ~ILib() = default;
29
37             virtual bool isKeyPressed(Arcade::Inputs input) = 0;
38
46             virtual bool isKeyReleased(Arcade::Inputs input) = 0;
47
53             virtual bool isWindowClosed(void) = 0;
54
58             virtual void updateEvent(void) = 0;
59
63             virtual void createWindow(void) = 0;
64
68             virtual void closeWindow(void) = 0;
69
73             virtual void clearWindow(void) = 0;
74
78             virtual void renderWindow(void) = 0;
79
85             virtual void drawObjets(std::shared_ptr<Arcade::IObject> object) = 0;
94             virtual void drawShapes(Arcade::Shapes shape, Arcade::Colors color, std::pair<ssize_t,
       ssize_t> pos, std::pair<ssize_t, ssize_t> size) = 0;
95
101             virtual void drawText(std::shared_ptr<Arcade::Text> text) = 0;
102
111             virtual void drawText(std::string str, Arcade::Colors color, ssize_t size,
       std::pair<ssize_t, ssize_t> pos) = 0;
112
118             virtual void setScale(std::pair<ssize_t, ssize_t> scale) = 0;
119
125             virtual void setScale(ssize_t scale) = 0;
126
132             virtual void setSize(std::pair<ssize_t, ssize_t> size) = 0;
133
139             virtual std::pair<ssize_t, ssize_t> getScale(void) const = 0;
140
146             virtual std::pair<ssize_t, ssize_t> getSize(void) const = 0;
147     };
148 }
```

## 6.5 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 {
12     enum Shapes {
13         SQUARE,
14         CIRCLE,
15         TRIANGLE,
16         NO_SHAPE
17     };
19     enum Inputs {
20         IKEY_UP,
21         IKEY_DOWN,
22         IKEY_LEFT,
23         IKEY_RIGHT,
24         IKEY_SPACE,
25         IKEY_ENTER,
26         IKEY_BACKSPACE,
27         IKEY_TAB,
28         IKEY_SHIFT,
29         IKEY_CTRL,
30         IKEY_ALT,
31
32         /* Globals inputs */
33         IKEY_B, //Graphical library
34         IKEY_D, //Game library
35         IKEY_S, //QUIT
36         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      };
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  }
```

## 6.6 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 {
16      class AObject : virtual public Arcade::IObject {
17          public:
18              AObject(void) = default;
19              ~AObject() = default;
20
21              Arcade::Shapes getShape(void) const final override {return _shape;};
22              std::pair<ssize_t, ssize_t> getPosition(void) const final override {return _position;};
23              std::pair<ssize_t, ssize_t> getSize(void) const final override {return _size;};
24              Arcade::Colors getColor(void) const final override {return _color;};
25              std::string getFilePath(void) const final override {return _texture;};
26
27              void setShape(Arcade::Shapes shape) final override {_shape = shape;};
28              void setPosition(std::pair<ssize_t, ssize_t> position) final override {_position =
        position;};
29              void setSize(std::pair<ssize_t, ssize_t> size) final override {_size = size;};
30              void setColor(Arcade::Colors color) final override {_color = color;};
31              void setFilePath(std::string texture) final override {_texture = texture;};
32
33          protected:
35              Arcade::Shapes _shape;
37              std::pair<ssize_t, ssize_t> _position;
39              std::pair<ssize_t, ssize_t> _size;
41              Arcade::Colors _color;
43              std::string _texture;
44      };
45  }
```

## 6.7 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
18     class IObject {
19         public:
24             IObject(void) = default;
25
29             virtual ~IObject() = default;
30
35             virtual Arcade::Shapes getShape(void) const = 0;
36
41             virtual void setShape(Arcade::Shapes) = 0;
42
47             virtual std::pair<ssize_t, ssize_t> getPosition(void) const = 0;
48
53             virtual void setPosition(std::pair<ssize_t, ssize_t>) = 0;
54
59             virtual std::pair<ssize_t, ssize_t> getSize(void) const = 0;
60
65             virtual void setSize(std::pair<ssize_t, ssize_t>) = 0;
66
71             virtual void setColor(Arcade::Colors) = 0;
72
77             virtual Arcade::Colors getColor(void) const = 0;
78
83             virtual std::string getFilePath(void) const = 0;
84
89             virtual void setFilePath(std::string) = 0;
90     };
91 };
```

## 6.8 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
14 namespace Arcade {
15
19     class Text {
20         public:
26             Text(void);
27
36             Text(std::string text, std::pair<ssize_t, ssize_t> pos, Arcade::Colors color);
37
41             ~Text() = default;
42
47             std::string getText(void) const { return _text;};
48
53             void setText(std::string) { _text = _text; };
54
59             std::pair<ssize_t, ssize_t> getPosition(void) const { return _position; };
60
65             void setPosition(std::pair<ssize_t, ssize_t>) { _position = _position; };
66
71             void setColor(Arcade::Colors) { _color = _color; };
72
77             Arcade::Colors getColor(void) const { return _color; };
78
79         private:
81             std::string _text;
83             std::pair<ssize_t, ssize_t> _position;
```

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