Indie Studio

Generated by Doxygen 1.8.13

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

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

Chapter 4

Class Documentation

4.1 Model3d::Animation Struct Reference

Represent a model's animation.

```
#include <Model3d.hh>
```

4.1.1 Detailed Description

Represent a model's animation.

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

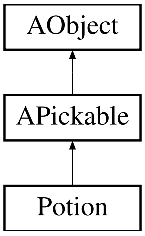
- inc/graphics/widgets/Model3d.hh

4.2 AObject Class Reference

An AObject instance has his own Model3d attribute, and can modify it..

```
#include <AObject.hh>
```

Inheritance diagram for AObject:



Public Member Functions

- AObject (std::shared_ptr< Model3d >, indie::objectType=indie::objectType::UNKNOW, irr::u32 id=0)

  Constructor Build the class first param is a shared_ptr on a Model3d, second the type of object (CHEST, POTION, ...), third his id.
- ∼AObject ()

  Destructor.
- indie::objectType getType () const

  getType
- std::shared_ptr< Model3d > getModel () const

  getModel
- void setPosition (Vector3d)

  setPosition
- void setRotation (Vector3d)

  setRotation
- void print () const

  print
- irr::u32 getId () const

  getId

4.2.1 Detailed Description

An AObject instance has his own Model3d attribute, and can modify it..

4.2.2 Constructor & Destructor Documentation

4.2.2.1 ∼AObject()

```
AObject::∼AObject ( )
```

Destructor.

Destroy the class

4.2.3 Member Function Documentation

4.2.3.1 getId()

```
irr::u32 AObject::getId ( ) const
```

getId

return the id of the AObject

4.2.3.2 getModel()

```
std::shared_ptr< Model3d > AObject::getModel ( ) const
```

getModel

return the Model3d of the AObject

4.2.3.3 getType()

```
indie::objectType AObject::getType ( ) const
```

getType

return the type of the AObject

4.2.3.4 print()

```
void AObject::print ( ) const
```

print

print the object on window

4.2.3.5 setPosition()

```
void AObject::setPosition (
             Vector3d pos )
```

setPosition

change the vector position of the Model3d of the AObject

4.2.3.6 setRotation()

```
void AObject::setRotation (
             Vector3d rot )
```

setRotation

change the vector rotation of the Model3d of the AObject

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

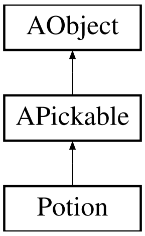
- inc/game/object/AObject.hh
- src/game/object/AObject.cpp

4.3 APickable Class Reference

Create a pickable AObject.

```
#include <APickable.hh>
```

Inheritance diagram for APickable:



Public Member Functions

- APickable (indie::objectType type)

  Constructor.
- ∼APickable ()

  Destructor.

4.3.1 Detailed Description

Create a pickable AObject.

4.3.2 Constructor & Destructor Documentation

4.3.2.1 APickable()

```
APickable::APickable (
            indie::objectType type )
```

Constructor.

Build the class

4.3.2.2 ∼APickable()

```
APickable::∼APickable ( )
```

Destructor.

Destroy the class

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

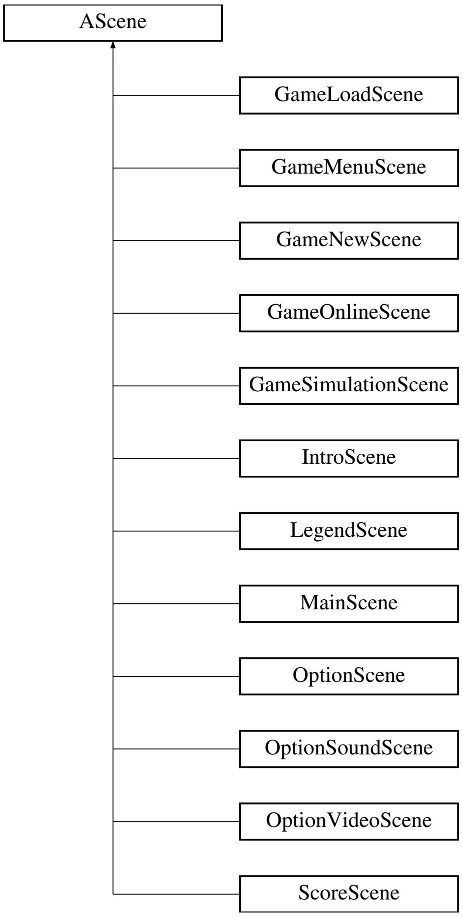
- inc/game/object/APickable.hh
- src/game/object/APickable.cpp

4.4 AScene Class Reference

Represent the window's scene that will print some widget/object of the scene : buttons, images, etc...

```
#include <AScene.hh>
```

Inheritance diagram for AScene:



Public Member Functions

- AScene ()

  Constructor.
- ∼AScene ()

  Destructor.
- void print (Window ∗win)

  Print all the scene's widgets into the Window.
- virtual void updateCamera (Window ∗win)

  Print all the scene's widgets into the Window.
- void addWidget (const std::shared_ptr< IWidget > &widget, Int id=-1)

  Allow the user to add texts, images, buttons, checkboxs or 3d models into the scene.
- void delWidget (Int id)

  Allow the user to delete texts, images, buttons, checkboxs or 3d models into the scene.
- virtual irr::IEventReceiver ∗ getEventReceiver () const =0

  Allow the user to implemente the event callback class for each scene.

4.4.1 Detailed Description

Represent the window's scene that will print some widget/object of the scene : buttons, images, etc...

4.4.2 Constructor & Destructor Documentation

4.4.2.1 AScene()

```
AScene::AScene ( )
```

Constructor.

Build the class.

Parameters

None	

4.4.2.2 ∼AScene()

```
AScene::∼AScene ( )
```

Destructor.

Destroy the class.

4.4.3 Member Function Documentation

4.4.3.1 addWidget()

```
void AScene::addWidget (
             const std::shared_ptr< IWidget > & widget,
             Int id = -1 )
```

Allow the user to add texts, images, buttons, checkboxs or 3d models into the scene.

Parameters

'widget'	the widget to add in the scene.

4.4.3.2 delWidget()

```
void AScene::delWidget (
            Int id )
```

Allow the user to delete texts, images, buttons, checkboxs or 3d models into the scene.

Example: scene->addWidget(widget, 2); scene->delWidget(2);

Parameters

'id'	the widget's id to del in the scene.

4.4.3.3 getEventReceiver()

```
virtual irr::IEventReceiver* AScene::getEventReceiver ( ) const  [pure virtual]
```

Allow the user to implemente the event callback class for each scene.

Parameters

None.	

Returns

The events' receiver of the scene.

Implemented in MainScene, GameOnlineScene, IntroScene, LegendScene, OptionScene, OptionVideoScene, GameSimulationScene, OptionSoundScene, GameLoadScene, GameMenuScene, GameNewScene, and ScoreScene.

4.4.3.4 print()

```
void AScene::print (
            Window * win )
```

Print all the scene's widgets into the Window.

Parameters

'win'	the window where the scene will be print.

4.4.3.5 updateCamera()

```
virtual void AScene::updateCamera (
            Window * win ) [inline], [virtual]
```

Print all the scene's widgets into the Window.

Parameters

'win'	the window where the scene will be print.

Reimplemented in GameSimulationScene.

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

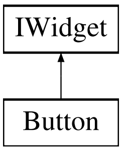
- inc/graphics/AScene.hh
- src/graphics/AScene.cpp

4.5 Button Class Reference

Represent a button in the user interface.

```
#include <Button.hh>
```

Inheritance diagram for Button:



Public Member Functions

- Button (const Rect &rect, const wchar_t *text, const wchar_t *tooltext, enum indie::GUIButtonId id, const wchar_t *background=L"assets/menus/transparent.png")

    Constructor.
- ∼Button ()

    Destructor.
- void print (Window *win)

    Print the button into the window.

4.5.1 Detailed Description

Represent a button in the user interface.

4.5.2 Constructor & Destructor Documentation

4.5.2.1 Button()

```
Button::Button (
            const Rect & rect,
            const wchar_t * text,
            const wchar_t * tooltext,
            enum indie::GUIButtonId id,
            const wchar_t * background = L"assets/menus/transparent.png" )
```

Constructor.

Build the class.

Parameters

1)	'rect' the rectangle who represent the button (x1, y1, x2, y2). 2) 'background' the name of the image, that will be put on the background of the button. 3) 'text' the text that will be write on the button. 3) 'tooltext' the text displayed in the tooltip. 4) 'id' the id throw the user click on the button .

4.5.2.2 ∼Button()

```
Button::∼Button ( )
```

Destructor.

Destroy the class.

4.5.3 Member Function Documentation

4.5.3.1 print()

```
void Button::print (
            Window * win )  [virtual]
```

Print the button into the window.

Parameters

'window'	the window where the button will be print.

Implements IWidget.

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

- inc/graphics/widgets/Button.hh
- src/graphics/widgets/Button.cpp

4.6 Camera Class Reference

where cameras behaviour is defined

```
#include <Camera.hh>
```

Public Member Functions

- Camera (irr::scene::ICameraSceneNode ∗const cam)

  instanciate Camera
- ∼Camera ()

  Destructor...
- void changePosition (irr::core::vector3df move)

  change absolute camera position
- void changeSight (irr::core::vector3df newTarget)

  change camera sight

4.6.1 Detailed Description

where cameras behaviour is defined

4.6.2 Constructor & Destructor Documentation

4.6.2.1 Camera()

```
Camera::Camera (
          irr::scene::ICameraSceneNode *const cam )
```

instanciate Camera

Parameters

cam	: the irrlicht static camera

4.6.3 Member Function Documentation

4.6.3.1 changePosition()

```
void Camera::changePosition (
             irr::core::vector3df move )
```

change absolute camera position

Parameters

move	: irrlicht 3D vector to positions camera

4.6.3.2 changeSight()

```
void Camera::changeSight (
             irr::core::vector3df newTarget )
```

change camera sight

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

- inc/graphics/Camera.hh
- src/graphics/Camera.cpp

4.7 ConfigManager Class Reference

This class contain all the methode to manipulate the configuration.

```
#include <ConfigManager.hh>
```

Public Member Functions

- ConfigManager ()

  Constructor.
- void loadConfig (std::string)

  Load the configuration.
- void saveConfig () const

  Savethe object in the file.
- void createFile (std::string)

  create an empty configuration file

Public Attributes

- config::Video videoConfig

  The object videoConfig is a structure that contain all the setting for the video.
- config::Sound sound

  The object sound is a structure that constain all the setting for the sound.
- config::Control control

  The structure control give all the informations about the control keys.

4.7.1 Detailed Description

This class contain all the methode to manipulate the configuration.

4.7.2 Constructor & Destructor Documentation

4.7.2.1 ConfigManager()

```
ConfigManager::ConfigManager ( )
```

Constructor.

Build the class

Parameters

nothing	

4.7.3 Member Function Documentation

4.7.3.1 createFile()

```
void ConfigManager::createFile (
             std::string fileName )
```

create an empty configuration file

Parameters

path	to the new configuration file

4.7.3.2 loadConfig()

```
void ConfigManager::loadConfig (
             std::string fileName )
```

Load the configuration.

Parameters

path	to the configuration file

Returns

void

4.7.4 Member Data Documentation

4.7.4.1 sound

```
config::Sound ConfigManager::sound
```

The object sound is a structure that constain all the setting for the sound.

The sructure is on the config namespace

4.7.4.2 videoConfig

```
config::Video ConfigManager::videoConfig
```

The object videoConfig is a structure that contain all the setting for the video.

The sructure is on the config namespace

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

- inc/managers/ConfigManager.hh
- src/managers/ConfigManager.cpp

4.8 CoreManager Class Reference

CoreManager is the main manager who regroups all the other manager.

```
#include <CoreManager.hh>
```

Public Member Functions

- CoreManager ()
  Constructor.
- ∼CoreManager ()
  Destructor.

4.8.1 Detailed Description

CoreManager is the main manager who regroups all the other manager.

4.8.2 Constructor & Destructor Documentation

4.8.2.1 CoreManager()

```
CoreManager::CoreManager ( )
```

Constructor.

Build the class.

Parameters

None	

4.8.2.2 ∼CoreManager()

```
CoreManager::∼CoreManager ( )
```

Destructor.

Destroy the class.

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

- inc/managers/CoreManager.hh
- src/managers/CoreManager.cpp

4.9 error::Exception Class Reference

This class allow to throw some execption and handle it finely.

```
#include <Exception.hh>
```

Inherits exception.

Public Member Functions

- Exception (const std::string &msg, const error::severity &severity, const char ∗file, const char ∗func, int line)

  /brief Constructor : Build the class.
- ∼Exception () throw ()

  /brief Destructor : Destroy the class.
- const error::severity & getSeverity () const

  /brief Give an access on the error's severity.
- const char ∗ what () const throw ()

  /brief Describe the error.

4.9.1 Detailed Description

This class allow to throw some execption and handle it finely.

4.9.2 Constructor & Destructor Documentation

4.9.2.1 Exception()

```
error::Exception::Exception (
            const std::string & msg,
            const error::severity & severity,
            const char ∗ file,
            const char ∗ func,
            int line )
```

/brief Constructor : Build the class.

/param This function take 5 parameters : 1) 'file' is the file where the error occurred. 2) 'func' is the func where the error occurred. 3) 'line' is the line where the error occurred. 4) 'msg' is the message who describe the error. 5) 'severity' is the error's severity.

/return None.

4.9.2.2 ∼Exception()

```
error::Exception::∼Exception ( ) throw ) [inline]
```

/brief Destructor : Destroy the class.

/param None.

/return None.

4.9.3 Member Function Documentation

4.9.3.1 getSeverity()

```
const error::severity & error::Exception::getSeverity ( ) const
```

/brief Give an access on the error's severity.

/param None.

/return None.

4.9.3.2 what()

```
const char * error::Exception::what ( ) const throw )
```

/brief Describe the error.

/param None.

/return the error's message.

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

- inc/exception/Exception.hh
- src/exception/Exception.cpp

4.10 Game Class Reference

The Game class.

```
#include <Game.hh>
```

Public Member Functions

- Game ()

  Constructor.
- ~Game ()

  Destructor.
- const std::shared_ptr< Level > & getLevel () const

  Return the level's attribute.
- void setLevel (const std::shared_ptr< Level > &level)

  Change the current level.

4.10.1 Detailed Description

The Game class.

4.10.2 Constructor & Destructor Documentation

4.10.2.1 Game()

```
Game::Game ( )
```

Constructor.

Build the class.

Parameters

None	

4.10.2.2 ∼Game()

```
Game::∼Game ( )
```

Destructor.

Destroy the class.

Parameters

None	

4.10.3 Member Function Documentation

4.10.3.1 getLevel()

```
const std::shared_ptr< Level > & Game::getLevel ( ) const
```

Return the level's attribute.

Parameters

None	

Returns

the current level.

4.10.3.2 setLevel()

```
void Game::setLevel (
            const std::shared_ptr< Level > & level )
```

Change the current level.

Parameters

The	new level.

Returns

None.

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

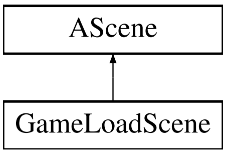
- inc/game/Game.hh
- src/game/Game.cpp

4.11 GameLoadScene Class Reference

Button interface of GameLoad scene.

```
#include <GameLoadScene.hh>
```

Inheritance diagram for GameLoadScene:



Public Member Functions

- GameLoadScene ()

  Constructor.
- ∼GameLoadScene ()

  Destructor.
- irr::IEventReceiver ∗ getEventReceiver () const

  Allow the user to implemente the event callback class for each scene.

4.11.1 Detailed Description

Button interface of GameLoad scene.

4.11.2 Constructor & Destructor Documentation

4.11.2.1 GameLoadScene()

```
GameLoadScene::GameLoadScene ( )
```

Constructor.

Build the class.

4.11.2.2 ∼GameLoadScene()

```
GameLoadScene::∼GameLoadScene ( )
```

Destructor.

Destroy the class.

4.11.3 Member Function Documentation

4.11.3.1 getEventReceiver()

```
irr::IEventReceiver * GameLoadScene::getEventReceiver ( ) const  [virtual]
```

Allow the user to implemente the event callback class for each scene.

Return the event.

Implements AScene.

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

- inc/graphics/scenes/GameLoadScene.hh
- src/graphics/scenes/GameLoadScene.cpp

4.12 GameMenuReceiver Class Reference

Manager of event of game scene.

```
#include <GameMenuReceiver.hh>
```

Inherits IEventReceiver.

Public Member Functions

- GameMenuReceiver ()

  Constructor.
- ∼GameMenuReceiver ()

  Destructor.
- bool OnEvent (const irr::SEvent &event)

  Event manager.

4.12.1 Detailed Description

Manager of event of game scene.

4.12.2 Constructor & Destructor Documentation

4.12.2.1 GameMenuReceiver()

```
GameMenuReceiver::GameMenuReceiver ( )
```

Constructor.

Build the class.

4.12.2.2 ∼GameMenuReceiver()

```
GameMenuReceiver::∼GameMenuReceiver ( )
```

Destructor.

Destroy the class.

4.12.3 Member Function Documentation

4.12.3.1 OnEvent()

```
bool GameMenuReceiver::OnEvent (
            const irr::SEvent & event )
```

Event manager.

Return true if he know the event, false if not.

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

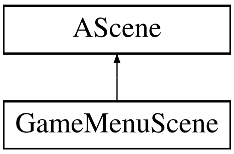
- inc/graphics/scenes/events/GameMenuReceiver.hh
- src/graphics/scenes/events/GameMenuReceiver.cpp

4.13 GameMenuScene Class Reference

Button interface of menu setting scene.

```
#include <GameMenuScene.hh>
```

Inheritance diagram for GameMenuScene:



Public Member Functions

- GameMenuScene ()

  Constructor.
- ∼GameMenuScene ()

  Destructor.
- irr::IEventReceiver ∗ getEventReceiver () const

  Allow the user to implemente the event callback class for each scene.

4.13.1 Detailed Description

Button interface of menu setting scene.

4.13.2 Constructor & Destructor Documentation

4.13.2.1 GameMenuScene()

```
GameMenuScene::GameMenuScene ( )
```

Constructor.

Build the class.

4.13.2.2 ∼GameMenuScene()

```
GameMenuScene::∼GameMenuScene ( )
```

Destructor.

Destroy the class.

4.13.3 Member Function Documentation

4.13.3.1 getEventReceiver()

```
irr::IEventReceiver * GameMenuScene::getEventReceiver ( ) const  [virtual]
```

Allow the user to implemente the event callback class for each scene.

Return the event.

Implements AScene.

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

- inc/graphics/scenes/GameMenuScene.hh
- src/graphics/scenes/GameMenuScene.cpp

4.14 GameNewReceiver Class Reference

Manager of event of new game scene.

```
#include <GameNewReceiver.hh>
```

Inherits IEventReceiver.

Public Member Functions

- GameNewReceiver (const GameNewScene ∗parent)

  Constructor.
- ∼GameNewReceiver ()

  Destructor.
- bool OnEvent (const irr::SEvent &event)

  Allow the user to handle envent into a scene.

4.14.1 Detailed Description

Manager of event of new game scene.

4.14.2 Constructor & Destructor Documentation

4.14.2.1 GameNewReceiver()

```
GameNewReceiver::GameNewReceiver (
            const GameNewScene * parent )
```

Constructor.

Build the class.

4.14.2.2 ∼GameNewReceiver()

```
GameNewReceiver::∼GameNewReceiver ( )
```

Destructor.

Destroy the class.

4.14.3 Member Function Documentation

4.14.3.1 OnEvent()

```
bool GameNewReceiver::OnEvent (
            const irr::SEvent & event )
```

Allow the user to handle envent into a scene.

Return true if he know the event, false if not.

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

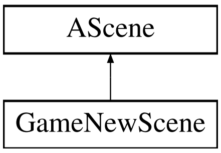
- inc/graphics/scenes/events/GameNewReceiver.hh
- src/graphics/scenes/events/GameNewReceiver.cpp

4.15 GameNewScene Class Reference

Button interface of new game setting scene.

`#include <GameNewScene.hh>`

Inheritance diagram for GameNewScene:



Public Member Functions

- GameNewScene ()

  Constructor.
- ∼GameNewScene ()

  Destructor.
- irr::IEventReceiver ∗ getEventReceiver () const

  Allow the user to implemente the event callback class for each scene.
- const std::vector< std::shared_ptr< AHero > > & getPlayers () const

  Return the players table.

4.15.1 Detailed Description

Button interface of new game setting scene.

4.15.2 Constructor & Destructor Documentation

4.15.2.1 GameNewScene()

```
GameNewScene::GameNewScene ( )
```

Constructor.

Build the class.

4.15.2.2 ∼GameNewScene()

```
GameNewScene::∼GameNewScene ( )
```

Destructor.

Destroy the class.

4.15.3 Member Function Documentation

4.15.3.1 getEventReceiver()

```
irr::IEventReceiver ∗ GameNewScene::getEventReceiver ( ) const [virtual]
```

Allow the user to implemente the event callback class for each scene.

Return the event.

Implements AScene.

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

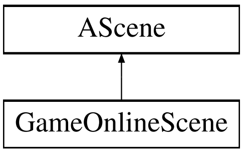
- inc/graphics/scenes/GameNewScene.hh
- src/graphics/scenes/GameNewScene.cpp

4.16 GameOnlineScene Class Reference

Button interface of GameOnlinescene.

```
#include <GameOnlineScene.hh>
```

Inheritance diagram for GameOnlineScene:



Public Member Functions

- GameOnlineScene ()

  Constructor.
- ∼GameOnlineScene ()

  Destructor.
- irr::IEventReceiver ∗ getEventReceiver () const

  Event getter.

4.16.1 Detailed Description

Button interface of GameOnlinescene.

4.16.2 Constructor & Destructor Documentation

4.16.2.1 GameOnlineScene()

```
GameOnlineScene::GameOnlineScene ( )
```

Constructor.

Build the class.

4.16.2.2 ∼GameOnlineScene()

```
GameOnlineScene::∼GameOnlineScene ( )
```

Destructor.

Destroy the class.

4.16.3 Member Function Documentation

4.16.3.1 getEventReceiver()

```
irr::IEventReceiver * GameOnlineScene::getEventReceiver ( ) const  [virtual]
```

Event getter.

Return the event.

Implements AScene.

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

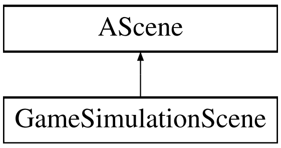
- inc/graphics/scenes/GameOnlineScene.hh
- src/graphics/scenes/GameOnlineScene.cpp

4.17 GameSimulationScene Class Reference

Button interface of GameSimulationscene.

```
#include <GameSimulationScene.hh>
```

Inheritance diagram for GameSimulationScene:



Public Member Functions

- GameSimulationScene (const std::shared_ptr< Level > &level)

  Constructor.
- ∼GameSimulationScene ()

  Destructor.
- void updateCamera (Window ∗win)

  Update camera's position.
- irr::IEventReceiver ∗ getEventReceiver () const

  Allow the user to implemente the event callback class for each scene.

4.17.1 Detailed Description

Button interface of GameSimulationscene.

4.17.2 Constructor & Destructor Documentation

4.17.2.1 GameSimulationScene()

```
GameSimulationScene::GameSimulationScene (
            const std::shared_ptr< Level > & level )
```

Constructor.

Build the class.

4.17.2.2 ∼GameSimulationScene()

```
GameSimulationScene::∼GameSimulationScene ( )
```

Destructor.

Destroy the class.

4.17.3 Member Function Documentation

4.17.3.1 getEventReceiver()

```
irr::IEventReceiver * GameSimulationScene::getEventReceiver ( ) const  [virtual]
```

Allow the user to implemente the event callback class for each scene.

Return the event.

Implements AScene.

4.17.3.2 updateCamera()

```
void GameSimulationScene::updateCamera (
            Window * win )  [virtual]
```

Update camera's position.

Parameters

The	window who contain the camera.

Reimplemented from AScene.

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

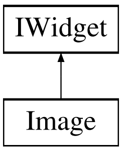
- inc/graphics/scenes/GameSimulationScene.hh
- src/graphics/scenes/GameSimulationScene.cpp

4.18 Image Class Reference

Represent a image in the user interface.

```
#include <Image.hh>
```

Inheritance diagram for Image:



Public Member Functions

- Image (const String &name, const Position2d &pos)

  Constructor.
- ∼Image ()

  Destructor.
- void print (Window ∗win)

  Print the image into the window.

4.18.1 Detailed Description

Represent a image in the user interface.

4.18.2 Constructor & Destructor Documentation

4.18.2.1 Image()

```
Image::Image (
            const String & name,
            const Position2d & pos )
```

Constructor.

Build the class.

Parameters

1)	'name' the name of the image that will be load and print. 2) 'pos' the position of the image where the image will be print.

4.18.2.2 ∼Image()

```
Image::∼Image ( )
```

Destructor.

Destroy the class.

4.18.3 Member Function Documentation

4.18.3.1 print()

```
void Image::print (
            Window * win )  [virtual]
```

Print the image into the window.

Parameters

'window'	the window where the image will be print.

Implements IWidget.

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

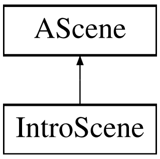
- inc/graphics/widgets/Image.hh
- src/graphics/widgets/Image.cpp

4.19 IntroScene Class Reference

Button interface of Introcene.

```
#include <IntroScene.hh>
```

Inheritance diagram for IntroScene:



Public Member Functions

- IntroScene ()

  Constructor.
- ∼IntroScene ()

  Destructor.
- irr::IEventReceiver ∗ getEventReceiver () const

  Event getter.

4.19.1 Detailed Description

Button interface of Introcene.

4.19.2 Constructor & Destructor Documentation

4.19.2.1 IntroScene()

```
IntroScene::IntroScene ( )
```

Constructor.

Build the class.

4.19.2.2 ∼IntroScene()

```
IntroScene::∼IntroScene ( )
```

Destructor.

Destroy the class.

4.19.3 Member Function Documentation

4.19.3.1 getEventReceiver()

```
irr::IEventReceiver ∗ IntroScene::getEventReceiver ( ) const  [virtual]
```

Event getter.

Return the event.

Implements AScene.

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

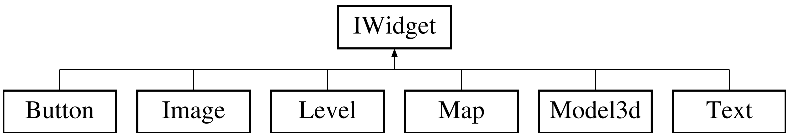
- inc/graphics/scenes/IntroScene.hh
- src/graphics/scenes/IntroScene.cpp

4.20 IWidget Class Reference

Represent a generic object in the user interface.

```
#include <IWidget.hh>
```

Inheritance diagram for IWidget:



Public Member Functions

- virtual ∼IWidget ()

    Destructor.
- virtual void print (Window ∗win)=0

    Print the object into the window.

4.20.1 Detailed Description

Represent a generic object in the user interface.

4.20.2 Constructor & Destructor Documentation

4.20.2.1 ∼IWidget()

```
virtual IWidget::∼IWidget ( )  [inline], [virtual]
```

Destructor.

Destroy the class.

4.20.3 Member Function Documentation

4.20.3.1 print()

```
virtual void IWidget::print (
            Window ∗ win )  [pure virtual]
```

Print the object into the window.

Parameters

'window'	the window where the object will be print.

Implemented in Level, Model3d, Map, Button, Text, and Image.

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

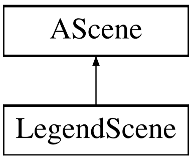
- inc/graphics/widgets/IWidget.hh

4.21 LegendScene Class Reference

Button interface of LegendScene.

```
#include <LegendScene.hh>
```

Inheritance diagram for LegendScene:



Public Member Functions

- LegendScene ()

  Constructor.
- ∼LegendScene ()

  Destructor.
- irr::IEventReceiver ∗ getEventReceiver () const

  Event getter.

4.21.1 Detailed Description

Button interface of LegendScene.

4.21.2 Constructor & Destructor Documentation

4.21.2.1 LegendScene()

```
LegendScene::LegendScene ( )
```

Constructor.

Build the class.

4.21.2.2 ∼LegendScene()

```
LegendScene::∼LegendScene ( )
```

Destructor.

Destroy the class.

4.21.3 Member Function Documentation

4.21.3.1 getEventReceiver()

```
irr::IEventReceiver ∗ LegendScene::getEventReceiver ( ) const  [virtual]
```

Event getter.

Return the event.

Implements AScene.

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

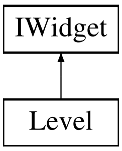
- inc/graphics/scenes/LegendScene.hh
- src/graphics/scenes/LegendScene.cpp

4.22 Level Class Reference

Represent a game's level.

```
#include <Level.hh>
```

Inheritance diagram for Level:



Public Member Functions

- Level (const WString &mapPath, std::vector< std::shared_ptr< AHero > > players=std::vector< std::shared_ptr< AHero > >(), std::vector< std::shared_ptr< AMonster > > monsters=std::vector< std::shared_ptr< AMonster > >(), std::vector< std::shared_ptr< APickable > > objects=std::vector< std::shared_ptr< APickable > >())

  Constructor.
- ∼Level ()

  Destructor.
- void movePlayer (const irr::u32 &playerID, const Vector3d &position)

  Change the player's position by adding the position passed as parameter.
- void loadLevel ()
- void print (Window ∗win)

  Print the object into the window.

4.22.1 Detailed Description

Represent a game's level.

4.22.2 Constructor & Destructor Documentation

4.22.2.1 Level()

```
Level::Level (
            const WString & mapPath,
            std::vector< std::shared_ptr< AHero > > players = std::vector< std::shared_↩
ptr<AHero> >(),
            std::vector< std::shared_ptr< AMonster > > monsters = std::vector< std::shared↩
_ptr<AMonster> >(),
            std::vector< std::shared_ptr< APickable > > objects = std::vector< std::shared↩
_ptr<APickable> >() )
```

Constructor.

Build the class

4.22.2.2 ∼Level()

```
Level::∼Level ( )
```

Destructor.

Destroy the class

4.22.3 Member Function Documentation

4.22.3.1 loadLevel()

```
void Level::loadLevel ( )
```

load level according to XML file xml is loaded thankfully to levelNumber Loading :

- map
- player
- monsters
- pickable objects

4.22.3.2 movePlayer()

```
void Level::movePlayer (
            const irr::u32 & playerID,
            const Vector3d & position )
```

Change the player's position by adding the position passed as parameter.

Parameters

1)	The player's id who will move. 2) The player's id who will move.

Returns

None.

4.22.3.3 print()

```
void Level::print (
            Window * win )  [virtual]
```

Print the object into the window.

Parameters

'window'	the window where the object will be print.

Implements IWidget.

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

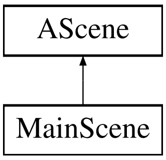
- inc/game/level/Level.hh
- src/game/level/Level.cpp

4.23 MainScene Class Reference

Button interface of main scene.

`#include <MainScene.hh>`

Inheritance diagram for MainScene:



Public Member Functions

- MainScene ()

    Constructor.
- ∼MainScene ()

    Destructor.
- irr::IEventReceiver ∗ getEventReceiver () const

    Event getter.

4.23.1 Detailed Description

Button interface of main scene.

4.23.2 Constructor & Destructor Documentation

4.23.2.1 MainScene()

```
MainScene::MainScene ( )
```

Constructor.

Build the class.

4.23.2.2 ∼MainScene()

```
MainScene::∼MainScene ( )
```

Destructor.

Destroy the class.

4.23.3 Member Function Documentation

4.23.3.1 getEventReceiver()

```
irr::IEventReceiver * MainScene::getEventReceiver ( ) const  [virtual]
```

Event getter.

Return the event.

Implements AScene.

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

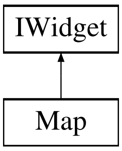
- inc/graphics/scenes/MainScene.hh
- src/graphics/scenes/MainScene.cpp

4.24 Map Class Reference

Represent a level's map.

```
#include <Map.hh>
```

Inheritance diagram for Map:



Public Member Functions

- Map (WString const mapName)

    Constructs the map with the related file.
- ∼Map ()

    Destructor...
- void print (Window ∗win)

    Print the map into the window, by adding some nodes into the scene manager.
- WString const getMapPath (void) const

    Accessor on map name.
- bool loadMap (Window ∗win) const

    Load map in global scene manager.
- bool loadSceneNodes (Window ∗win)

    Load all the nodes of the map.
- bool initCollisionResponseAnimator (Window ∗win)

    Initialize the collisions.

4.24.1 Detailed Description

Represent a level's map.

4.24.2 Constructor & Destructor Documentation

4.24.2.1 Map()

```
Map::Map (
            WString const mapName )
```

Constructs the map with the related file.

Parameters

mapName	: the related .irr map file

4.24.3 Member Function Documentation

4.24.3.1 loadMap()

```
bool Map::loadMap (
            Window * win ) const
```

Load map in global scene manager.

.irr files can store the whole scene graph including animators, materials and particle systems. And there is also the possibility to store arbitrary user data for every scene node in that file. this example simple, we are simply loading the scene here. See the documentation at ISceneManager::loadScene and ISceneManager::saveScene for more information.

4.24.3.2 loadSceneNodes()

```
bool Map::loadSceneNodes (
            Window * win )
```

Load all the nodes of the map.

We find all the nodes in the scene It would be possible to make a more informed decision about which nodes to perform collision checks on. We would such capture that information in the node name/id.

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

- inc/game/level/Map.hh
- src/game/level/Map.cpp

4.25 MapFactory Class Reference

Represent a maps' factory.

```
#include <MapFactory.hh>
```

4.25.1 Detailed Description

Represent a maps' factory.

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

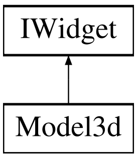
- inc/game/level/MapFactory.hh
- src/game/level/MapFactory.cpp

4.26 Model3d Class Reference

Represent a model 3d in the user interface.

```
#include <Model3d.hh>
```

Inheritance diagram for Model3d:



Classes

- struct Animation

  Represent a model's animation.

Public Member Functions

- Model3d (const String &fileXML, const Vector3d &position, const Vector3d &rotation, Float scale=1, size_t animationSpeed=4)

  Constructor.
- ∼Model3d ()

  Destructor.
- void print (Window ∗win)

  Print the model 3d into the window.
- void setCurrentAnim (const String &name)

  Allow to change the current animated mesh of the model.

4.26.1 Detailed Description

Represent a model 3d in the user interface.

4.26.2 Constructor & Destructor Documentation

4.26.2.1 Model3d()

```
Model3d::Model3d (
            const String & fileXML,
            const Vector3d & position,
            const Vector3d & rotation,
            Float scale = 1,
            size_t animationSpeed = 4 )
```

Constructor.

Build the class.

Parameters

1)	'fileXML' path to the xml file who describe the model. 2) 'position' the position in the space where the model will be print. 3) 'rotation' the rotation of the model in the space. 4) 'scale' an additionnal scale coefficient for the model.

4.26.2.2 ∼Model3d()

```
Model3d::∼Model3d ( )
```

Destructor.

Destroy the class.

4.26.3 Member Function Documentation

4.26.3.1 print()

```
void Model3d::print (
             Window * win )  [virtual]
```

Print the model 3d into the window.

Parameters

'window'	the window where the model 3d will be print.

Implements IWidget.

4.26.3.2 setCurrentAnim()

```
void Model3d::setCurrentAnim (
             const String & name )
```

Allow to change the current animated mesh of the model.

Parameters

'name'	of the animation that will be load.

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

- inc/graphics/widgets/Model3d.hh
- src/graphics/widgets/Model3d.cpp

4.27 ModelsManager Class Reference

The class to load all the information about a model.

```
#include <ModelsManager.hh>
```

Public Member Functions

- ModelsManager ()

  Constructor.
- bool loadModel (String, String)

  Allow you to load a model.
- const config::Model ∗ getModel (String) const

  Give you a pointeur to an object Model.

4.27.1 Detailed Description

The class to load all the information about a model.

4.27.2 Constructor & Destructor Documentation

4.27.2.1 ModelsManager()

```
ModelsManager::ModelsManager ( )
```

Constructor.

Parameters

nothing	

4.27.3 Member Function Documentation

4.27.3.1 getModel()

```
const config::Model ∗ ModelsManager::getModel (
            String toFind ) const
```

Give you a pointeur to an object Model.

Parameters

The	name previously given to the model

Returns

A pointer to the model

4.27.3.2 loadModel()

```
bool ModelsManager::loadModel (
             String path,
             String modelName )
```

Allow you to load a model.

Parameters

the	path to the xml configuration and a name

Returns

true is the model is load and false otherwise

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

- inc/managers/ModelsManager.hh
- src/managers/ModelsManager.cpp

4.28 OptionControl Class Reference

Button interface of OptionControlscene.

```
#include <OptionControlScene.hh>
```

4.28.1 Detailed Description

Button interface of OptionControlscene.

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

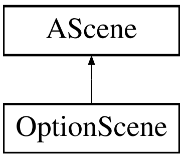
- inc/graphics/scenes/OptionControlScene.hh

4.29 OptionScene Class Reference

Button interface of option scene.

```
#include <OptionScene.hh>
```

Inheritance diagram for OptionScene:



Public Member Functions

- OptionScene ()

  Constructor.
- ∼OptionScene ()

  Destructor.
- irr::IEventReceiver ∗ getEventReceiver () const

  Event getter.

4.29.1 Detailed Description

Button interface of option scene.

4.29.2 Constructor & Destructor Documentation

4.29.2.1 OptionScene()

```
OptionScene::OptionScene ( )
```

Constructor.

Build the class.

4.29.2.2 ∼OptionScene()

```
OptionScene::∼OptionScene ( )
```

Destructor.

Destroy the class.

4.29.3 Member Function Documentation

4.29.3.1 getEventReceiver()

```
irr::IEventReceiver * OptionScene::getEventReceiver ( ) const  [virtual]
```

Event getter.

Return the event.

Implements AScene.

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

- inc/graphics/scenes/OptionScene.hh
- src/graphics/scenes/OptionScene.cpp

4.30 OptionSoundReceiver Class Reference

Manager of event of Sound scene setting.

```
#include <OptionSoundReceiver.hh>
```

Inherits IEventReceiver.

Public Member Functions

- OptionSoundReceiver (OptionSoundScene ∗)

  Constructor.
- ∼OptionSoundReceiver ()

  Destructor.
- bool OnEvent (const irr::SEvent &event)

  Allow the user to handle envent into a scene.

4.30.1 Detailed Description

Manager of event of Sound scene setting.

4.30.2 Constructor & Destructor Documentation

4.30.2.1 OptionSoundReceiver()

```
OptionSoundReceiver::OptionSoundReceiver (
            OptionSoundScene * scene )
```

Constructor.

Build the class.

4.30.2.2 ∼OptionSoundReceiver()

```
OptionSoundReceiver::∼OptionSoundReceiver ( )
```

Destructor.

Destroy the class.

4.30.3 Member Function Documentation

4.30.3.1 OnEvent()

```
bool OptionSoundReceiver::OnEvent (
            const irr::SEvent & event )
```

Allow the user to handle envent into a scene.

Return true if he know the event, false if not.

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

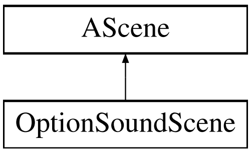
- inc/graphics/scenes/events/OptionSoundReceiver.hh
- src/graphics/scenes/events/OptionSoundReceiver.cpp

4.31 OptionSoundScene Class Reference

Button interface of sound setting scene.

```
#include <OptionSoundScene.hh>
```

Inheritance diagram for OptionSoundScene:



Public Member Functions

- OptionSoundScene ()

  Constructor.
- ∼OptionSoundScene ()

  Destructor.
- irr::IEventReceiver ∗ getEventReceiver () const

  Allow the user to implemente the event callback class for each scene.

4.31.1 Detailed Description

Button interface of sound setting scene.

4.31.2 Constructor & Destructor Documentation

4.31.2.1 OptionSoundScene()

```
OptionSoundScene::OptionSoundScene ( )
```

Constructor.

Build the class.

4.31.2.2 ∼OptionSoundScene()

```
OptionSoundScene::∼OptionSoundScene ( )
```

Destructor.

Destroy the class.

4.31.3 Member Function Documentation

4.31.3.1 getEventReceiver()

```
irr::IEventReceiver ∗ OptionSoundScene::getEventReceiver ( ) const  [virtual]
```

Allow the user to implemente the event callback class for each scene.

Return the event's receiver.

Implements AScene.

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

- inc/graphics/scenes/OptionSoundScene.hh
- src/graphics/scenes/OptionSoundScene.cpp

4.32 OptionVideoReceiver Class Reference

Manager of event of video setting scene.

```
#include <OptionVideoReceiver.hh>
```

Inherits IEventReceiver.

Public Member Functions

- OptionVideoReceiver ()

  Constructor.
- ~OptionVideoReceiver ()

  Destructor.
- bool OnEvent (const irr::SEvent &event)

  Event manager.

4.32.1 Detailed Description

Manager of event of video setting scene.

4.32.2 Constructor & Destructor Documentation

4.32.2.1 OptionVideoReceiver()

```
OptionVideoReceiver::OptionVideoReceiver ( )
```

Constructor.

Build the class.

4.32.2.2 ~OptionVideoReceiver()

```
OptionVideoReceiver::~OptionVideoReceiver ( )
```

Destructor.

Destroy the class.

4.32.3 Member Function Documentation

4.32.3.1 OnEvent()

```
bool OptionVideoReceiver::OnEvent (
            const irr::SEvent & event )
```

Event manager.

Return true if he know the event, false if not.

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

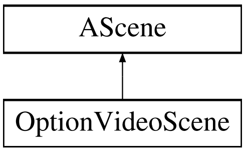
- inc/graphics/scenes/events/OptionVideoReceiver.hh
- src/graphics/scenes/events/OptionVideoReceiver.cpp

4.33 OptionVideoScene Class Reference

Button interface of video setting scene.

```
#include <OptionVideoScene.hh>
```

Inheritance diagram for OptionVideoScene:



Public Member Functions

- OptionVideoScene ()

  Constructor.
- ∼OptionVideoScene ()

  Destructor.
- irr::IEventReceiver ∗ getEventReceiver () const

  Event getter.

4.33.1 Detailed Description

Button interface of video setting scene.

4.33.2 Constructor & Destructor Documentation

4.33.2.1 OptionVideoScene()

```
OptionVideoScene::OptionVideoScene ( )
```

Constructor.

Build the class.

4.33.2.2 ∼OptionVideoScene()

```
OptionVideoScene::∼OptionVideoScene ( )
```

Destructor.

Destroy the class.

4.33.3 Member Function Documentation

4.33.3.1 getEventReceiver()

```
irr::IEventReceiver * OptionVideoScene::getEventReceiver ( ) const  [virtual]
```

Event getter.

Return the event.

Implements AScene.

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

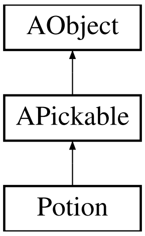
- inc/graphics/scenes/OptionVideoScene.hh
- src/graphics/scenes/OptionVideoScene.cpp

4.34 Potion Class Reference

create a APickable AObject key

```
#include <Potion.hh>
```

Inheritance diagram for Potion:



Public Member Functions

- Potion (irr::s32 regenPoint, indie::potionType type=indie::potionType::LIFE, irr::u32 id=0)

  Constructor.
- ∼Potion ()

  Destructor.

4.34.1 Detailed Description

create a APickable AObject key

4.34.2 Constructor & Destructor Documentation

4.34.2.1 Potion()

```
Potion::Potion (
            irr::s32 regenPoint,
            indie::potionType type = indie::potionType::LIFE,
            irr::u32 id = 0 )
```

Constructor.

Build the class, first param is number of regenPoint, second the type of potion (LIFE, MANA, POWER)

4.34.2.2 ∼Potion()

```
Potion::∼Potion ( )
```

Destructor.

Destroy a class

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

- inc/game/object/Potion.hh
- src/game/object/Potion.cpp

4.35 SaveManager Class Reference

The SaveManager class.

```
#include <SaveManager.hh>
```

Public Member Functions

- SaveManager ()

  The constructor.
- std::shared_ptr< Level > loadLevel (const String &levelName)

  Allow you to load a game previously save.
- void saveLevel (const std::shared_ptr< Level > &level, const String &fileName)

  Allow you to save the object Game passed as parameter during the construction.
- const std::shared_ptr< Game > & getGame () const

  Getter on the game's variable.

4.35.1 Detailed Description

The SaveManager class.

4.35.2 Member Function Documentation

4.35.2.1 loadLevel()

```
std::shared_ptr< Level > SaveManager::loadLevel (
            const String & levelName )
```

Allow you to load a game previously save.

Parameters

The	path to the game's file

Returns

A pointeur to a Game object

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

- inc/managers/SaveManager.hh
- src/managers/SaveManager.cpp

4.36 ScoreManager Class Reference

The ScoreManager class.

```
#include <ScoreManager.hh>
```

Public Member Functions

- ScoreManager ()

  The constructor.
- void loadScores (std::string)

  Allow you to load a score sheet (from a previous session)
- void saveScores () const

  Allow you to save the score contained in the object.
- void createFile (std::string)

  Create an empty score file.
- void addScore (std::string, std::size_t)

  Add a score.
- const std::vector< config::Score > & getTopScores ()

  Get the 10 best scores.
- std::size_t getPlayerScore (std::string) const

  Get score for a specifique player.
- void changeScore (std::string, std::size_t)

  Change score for a specifique player.

4.36.1 Detailed Description

The ScoreManager class.

4.36.2 Constructor & Destructor Documentation

4.36.2.1 ScoreManager()

```
ScoreManager::ScoreManager ( )
```

The constructor.

Parameters

Nothing	

4.36.3 Member Function Documentation

4.36.3.1 addScore()

```
void ScoreManager::addScore (
            std::string playerName,
            std::size_t playerScore )
```

Add a score.

Parameters

1.	Player name 2. Is score

4.36.3.2 changeScore()

```
void ScoreManager::changeScore (
             std::string ,
             std::size_t  )
```

Change score for a specifique player.

Parameters

1.	Player name 2. New score

4.36.3.3 createFile()

```
void ScoreManager::createFile (
             std::string fileName )
```

Create an empty score file.

Parameters

Path	to the score file

4.36.3.4 getPlayerScore()

```
std::size_t ScoreManager::getPlayerScore (
             std::string name ) const
```

Get score for a specifique player.

Parameters

Player	name

4.36.3.5 loadScores()

```
void ScoreManager::loadScores (
             std::string fileName )
```

Allow you to load a score sheet (from a previous session)

Parameters

The	path to the score file

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

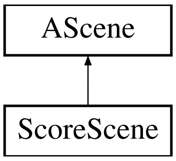
- inc/managers/ScoreManager.hh
- src/managers/ScoreManager.cpp

4.37 ScoreScene Class Reference

Button interface of ScoreScene.

`#include <ScoreScene.hh>`

Inheritance diagram for ScoreScene:



Public Member Functions

- ScoreScene ()
  Constructor.
- ∼ScoreScene ()
  Destructor.
- irr::IEventReceiver ∗ getEventReceiver () const
  Event getter.
- WString to_WString (std::string) const
  to_Wstring
- WString getScores () const
  Event getter.

4.37.1 Detailed Description

Button interface of ScoreScene.

4.37.2 Constructor & Destructor Documentation

4.37.2.1 ScoreScene()

```
ScoreScene::ScoreScene ( )
```

Constructor.

Build the class.

4.37.2.2 ∼ScoreScene()

```
ScoreScene::∼ScoreScene ( )
```

Destructor.

Destroy the class.

4.37.3 Member Function Documentation

4.37.3.1 getEventReceiver()

```
irr::IEventReceiver * ScoreScene::getEventReceiver ( ) const  [virtual]
```

Event getter.

Return the event.

Implements AScene.

4.37.3.2 getScores()

```
WString ScoreScene::getScores ( ) const
```

Event getter.

Return the Score.

4.37.3.3 to_WString()

```
WString ScoreScene::to_WString (
            std::string txt ) const
```

to_Wstring

exchange a string to a Wstring for irrlicht

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

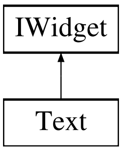
- inc/graphics/scenes/ScoreScene.hh
- src/graphics/scenes/ScoreScene.cpp

4.38 Text Class Reference

Inherited from a IWidget.

`#include <Text.hh>`

Inheritance diagram for Text:



Public Member Functions

- Text (WString text, Rect pos, std::string fontName="chow_fun", std::size_t fontSize=96)

  Constructor.
- void print (Window ∗win)

  Display the text box on the screen.

4.38.1 Detailed Description

Inherited from a IWidget.

4.38.2 Constructor & Destructor Documentation

4.38.2.1 Text()

```
Text::Text (
            WString text,
            Rect pos,
            std::string fontName = "chow_fun",
            std::size_t fontSize = 96 )
```

Constructor.

Parameters

text	The data
pos	The position and the size of the text box
fontName	The font name to be used
fontSize	The size of the font (8, 9, 10, 11, 12, 14, 18, 24, 30, 36, 48, 60, 72, 96)

4.38.3 Member Function Documentation

4.38.3.1 print()

```
void Text::print (
            Window * win )  [virtual]
```

Display the text box on the screen.

Parameters

win	The window

Implements IWidget.

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

- inc/graphics/widgets/Text.hh
- src/graphics/widgets/Text.cpp

4.39 Window Class Reference

Represent the window that will print some scene (view of the application).

`#include <Window.hh>`

Public Member Functions

- Window (const wchar_t *windowName)

  Constructor.
- ∼Window ()

  Destructor.
- void addScene (const String &name, const std::shared_ptr< AScene > &scene)

  Allow the user to add a scene in the window.
- void printScene (const String &name)

  Allow the user to print a scene in the window.

- void deleteScene (const String &name)
  Allow the user ti delete a scene in the window.
- void setCurrentScene (const String &sceneName)
  Set the current scene that will be print.
- void changeScene ()
  Apply the setCurrentScene's changes.
- void close ()
  Close the window.
- bool isOpen () const
  Getter on the open's statement of the window.
- void loop (const String &sceneName)
  Keep the window open util the end of the program's execution.

4.39.1 Detailed Description

Represent the window that will print some scene (view of the application).

4.39.2 Constructor & Destructor Documentation

4.39.2.1 Window()

```
Window::Window (
            const wchar_t * windowName )
```

Constructor.

Build the class.

Parameters

'windowName'	the window name that will be print on the top of the window.

4.39.2.2 ∼Window()

```
Window::∼Window ( )
```

Destructor.

Destroy the class.

4.39.3 Member Function Documentation

4.39.3.1 addScene()

```
void Window::addScene (
             const String & name,
             const std::shared_ptr< AScene > & scene )
```

Allow the user to add a scene in the window.

Parameters

name	The name of the scene that will be add.
scene	Reference to a smart pointer on the scene that will be add.

4.39.3.2 deleteScene()

```
void Window::deleteScene (
             const String & name )
```

Allow the user ti delete a scene in the window.

Parameters

'name'	the name of the scene that will be delete.

4.39.3.3 isOpen()

```
bool Window::isOpen ( ) const
```

Getter on the open's statement of the window.

Returns

True if the window is open, else false.

4.39.3.4 loop()

```
void Window::loop (
             const String & sceneName )
```

Keep the window open util the end of the program's execution.

Parameters

'sceneName'	the scene's name that will be launch at the begin of the program's execution.

4.39.3.5 printScene()

```
void Window::printScene (
             const String & name )
```

Allow the user to print a scene in the window.

Parameters

'name'	the name of the scene that will be print.

4.39.3.6 setCurrentScene()

```
void Window::setCurrentScene (
             const String & sceneName )
```

Set the current scene that will be print.

Parameters

'sceneName'	the name of the scene that will be set as the current scene.

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

- inc/graphics/Window.hh
- src/graphics/Window.cpp

Chapter 5

File Documentation

5.1 inc/exception/Exception.hh File Reference

Exception class.

```
#include <exception>
#include <sstream>
#include <string>
```

Classes

- class error::Exception

  This class allow to throw some execption and handle it finely.

5.1.1 Detailed Description

Exception class.

Author

Théophile Champion

Version

1.0

5.2 inc/game/Game.hh File Reference

Contain all the information about the game.

```
#include <memory>
#include "Level.hh"
```

Classes

- class Game

  The Game class.

5.2.1 Detailed Description

Contain all the information about the game.

Author

Théophile Champion

Version

1.0

5.3 inc/game/level/Level.hh File Reference

Use to create a level of the game.

```
#include <irrlicht/irrlicht.h>
#include <vector>
#include <memory>
#include <string>
#include "MapFactory.hh"
#include "Map.hh"
```

Classes

- class Level

  Represent a game's level.

5.3.1 Detailed Description

Use to create a level of the game.

Author

Théophile Champion

Version

1.0

5.4 inc/game/level/Map.hh File Reference

Defines the map that is related to a level. The process to create a map is relatively simple. First of all, you've got to load a map with Map::loadMap() In a second time, you've got to retrieve all the nodes with a triangle selector container (meta) with loadSceneNodes Then, you can create collision response thanks to these nodes by calling initCollisionResponseAnimator() Finally, you can draw the map (that should be achieved with drawAll() bugs: If meshs/models arent loading properly. Double check the paths in the .irr file.

```
#include <iostream>
#include <string>
#include "graphics_header.hh"
#include "IWidget.hh"
```

Classes

- class Map

  Represent a level's map.

5.4.1 Detailed Description

Defines the map that is related to a level. The process to create a map is relatively simple. First of all, you've got to load a map with Map::loadMap() In a second time, you've got to retrieve all the nodes with a triangle selector container (meta) with loadSceneNodes Then, you can create collision response thanks to these nodes by calling initCollisionResponseAnimator() Finally, you can draw the map (that should be achieved with drawAll() bugs: If meshs/models arent loading properly. Double check the paths in the .irr file.

Author

Charles Paulet

Version

0.1

5.5 inc/game/level/MapFactory.hh File Reference

Maps' factory use to build the maps objects.

```
#include <memory>
#include "Map.hh"
```

Classes

- class MapFactory

  Represent a maps' factory.

5.5.1 Detailed Description

Maps' factory use to build the maps objects.

Author

Charles Paulet

Version

0.1

5.6 inc/game/object/AObject.hh File Reference

Use to create object.

```
#include <irrlicht/irrlicht.h>
#include "Model3d.hh"
```

Classes

- class AObject

  An AObject instance has his own Model3d attribute, and can modify it..

5.6.1 Detailed Description

Use to create object.

Author

Hugo Ailleres

Version

1.0

5.7 inc/game/object/APickable.hh File Reference

Use to create pickable object.

```
#include "AObject.hh"
```

Classes

- class APickable

  Create a pickable AObject.

5.7.1 Detailed Description

Use to create pickable object.

Use to create a key APickable AObject.

Author

Hugo Ailleres

Version

1.0

5.8 inc/graphics/AScene.hh File Reference

Define a window's scene.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include <vector>
#include "type_compatibility.hh"
```

Classes

- class AScene

  Represent the window's scene that will print some widget/object of the scene : buttons, images, etc...

5.8.1 Detailed Description

Define a window's scene.

Author

Théophile Champion

Version

1.0

5.9 inc/graphics/Camera.hh File Reference

Defines the camera, its type and behaviour bugs: Actually you have to give the constructor the camera. That may change soon with integration on indie_studio This way, cam will be created inside cosntructor with the g_Window->sceneManager that would be such accessible.

Classes

- class Camera

  where cameras behaviour is defined

5.9.1 Detailed Description

Defines the camera, its type and behaviour bugs: Actually you have to give the constructor the camera. That may change soon with integration on indie_studio This way, cam will be created inside cosntructor with the g_Window->sceneManager that would be such accessible.

Camera positionning may be set using Maths::getCentroid as position

Author

Charles Paulet

Version

0.1

5.10 inc/graphics/scenes/events/GameMenuReceiver.hh File Reference

Use to manage event of the game (new game / load) scene.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "AScene.hh"
```

Classes

- class GameMenuReceiver

  Manager of event of game scene.

5.10.1 Detailed Description

Use to manage event of the game (new game / load) scene.

Author

Leo PAOL

Version

1.0

5.11 inc/graphics/scenes/events/GameNewReceiver.hh File Reference

Use to manage event of the new game (character selection) scene.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "AScene.hh"
```

Classes

- class GameNewReceiver

  Manager of event of new game scene.

5.11.1 Detailed Description

Use to manage event of the new game (character selection) scene.

Author

Leo PAOL

Version

1.0

5.12 inc/graphics/scenes/events/OptionSoundReceiver.hh File Reference

Allow to manage event of Sound scene setting.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "AScene.hh"
```

Classes

- class OptionSoundReceiver

  Manager of event of Sound scene setting.

5.12.1 Detailed Description

Allow to manage event of Sound scene setting.

Author

Hugo Ailleres

Version

1.0

5.13 inc/graphics/scenes/events/OptionVideoReceiver.hh File Reference

Use to manage event of the video setting scene.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "AScene.hh"
```

Classes

- class OptionVideoReceiver

  Manager of event of video setting scene.

5.13.1 Detailed Description

Use to manage event of the video setting scene.

Author

Leo PAOL

Version

1.0

5.14 inc/graphics/scenes/GameLoadScene.hh File Reference

Use to display the gameload scene.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "graphics_header.hh"
```

Classes

- class GameLoadScene

  Button interface of GameLoad scene.

5.14.1 Detailed Description

Use to display the gameload scene.

Author

Clovis Peridy

Version

1.0

5.15 inc/graphics/scenes/GameMenuScene.hh File Reference

Use to choose between new game and load game.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "graphics_header.hh"
```

Classes

- class GameMenuScene

  Button interface of menu setting scene.

5.15.1 Detailed Description

Use to choose between new game and load game.

Author

Leo PAOL

Version

1.0

5.16 inc/graphics/scenes/GameNewScene.hh File Reference

Use to choose characters for each player.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "graphics_header.hh"
```

Classes

- class GameNewScene

  Button interface of new game setting scene.

5.16.1 Detailed Description

Use to choose characters for each player.

Author

Léo PAOL

Version

1.0

5.17 inc/graphics/scenes/GameOnlineScene.hh File Reference

Use to display the GameOnline scene.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "graphics_header.hh"
```

Classes

- class GameOnlineScene

  Button interface of GameOnlinescene.

5.17.1 Detailed Description

Use to display the GameOnline scene.

Author

Clovis Peridy

Version

1.0

5.18 inc/graphics/scenes/GameScene.hh File Reference

Use to display the Game scene.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "graphics_header.hh"
```

5.18.1 Detailed Description

Use to display the Game scene.

Author

Clovis Peridy

Version

1.0

5.19 inc/graphics/scenes/GameSimulationScene.hh File Reference

Use to display the GameOnline scene.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "graphics_header.hh"
```

Classes

- class GameSimulationScene

  Button interface of GameSimulationscene.

5.19.1 Detailed Description

Use to display the GameOnline scene.

Author

Clovis Peridy

Version

1.0

5.20 inc/graphics/scenes/IntroScene.hh File Reference

Use to display the IntroScene scene.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "graphics_header.hh"
```

Classes

- class IntroScene

  Button interface of Introcene.

5.20.1 Detailed Description

Use to display the IntroScene scene.

Author

Clovis Peridy

Version

1.0

5.21 inc/graphics/scenes/LegendScene.hh File Reference

Use to display the Legend scene.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "graphics_header.hh"
```

Classes

- class LegendScene

  Button interface of LegendScene.

5.21.1 Detailed Description

Use to display the Legend scene.

Author

Clovis Peridy

Version

1.0

5.22 inc/graphics/scenes/MainScene.hh File Reference

Main scene who lead to each menus.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "graphics_header.hh"
```

Classes

- class MainScene

  Button interface of main scene.

5.22.1 Detailed Description

Main scene who lead to each menus.

Author

Clovis Peridy

Version

1.0

5.23 inc/graphics/scenes/OptionControlScene.hh File Reference

Use to display the OptionControl scene.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "graphics_header.hh"
```

5.23.1 Detailed Description

Use to display the OptionControl scene.

Author

Clovis Peridy

Version

1.0

5.24 inc/graphics/scenes/OptionScene.hh File Reference

Use to display the option menu.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "graphics_header.hh"
```

Classes

- class OptionScene

  Button interface of option scene.

5.24.1 Detailed Description

Use to display the option menu.

Author

Clovis Peridy

Version

1.0

5.25 inc/graphics/scenes/OptionSoundScene.hh File Reference

Use to create the button interface of the sound setting scene.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "graphics_header.hh"
```

Classes

- class OptionSoundScene

  Button interface of sound setting scene.

5.25.1 Detailed Description

Use to create the button interface of the sound setting scene.

Author

Hugo Ailleres

Version

1.0

5.26 inc/graphics/scenes/OptionVideoScene.hh File Reference

Use to create the button interface of the video setting scene.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "graphics_header.hh"
```

Classes

- class OptionVideoScene

  Button interface of video setting scene.

5.26.1 Detailed Description

Use to create the button interface of the video setting scene.

Author

Leo PAOL

Version

1.0

5.27 inc/graphics/scenes/ScoreScene.hh File Reference

Use to display the Score scene.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "ScoreManager.hh"
#include "graphics_header.hh"
#include "Text.hh"
```

Classes

- class ScoreScene

  Button interface of ScoreScene.

5.27.1 Detailed Description

Use to display the Score scene.

Author

Clovis Peridy

Version

1.0

5.28 inc/graphics/widgets/Button.hh File Reference

Allow to add some button in the window.

```
#include "type_compatibility.hh"
#include "graphics_header.hh"
#include "GUIButtonId.hh"
```

Classes

- class Button

  Represent a button in the user interface.

5.28.1 Detailed Description

Allow to add some button in the window.

Author

Théophile Champion

Version

1.0

5.29 inc/graphics/widgets/Image.hh File Reference

Allow to add some image in the window.

```
#include <irrlicht/irrlicht.h>
#include "type_compatibility.hh"
#include "graphics_header.hh"
```

Classes

- class Image

  Represent a image in the user interface.

5.29.1 Detailed Description

Allow to add some image in the window.

Author

Théophile Champion

Version

1.0

5.30 inc/graphics/widgets/IWidget.hh File Reference

Give a interface for all the class that could be print on the window.

```
#include "graphics_header.hh"
```

Classes

- class IWidget

  Represent a generic object in the user interface.

5.30.1 Detailed Description

Give a interface for all the class that could be print on the window.

Author

Théophile Champion

Version

1.0

5.31 inc/graphics/widgets/Model3d.hh File Reference

Allow to add some model 3d in the window.

```
#include <vector>
#include <string>
#include "type_compatibility.hh"
#include "graphics_header.hh"
```

Classes

- class Model3d

  Represent a model 3d in the user interface.
- struct Model3d::Animation

  Represent a model's animation.

5.31.1 Detailed Description

Allow to add some model 3d in the window.

Author

Théophile Champion

Version

1.0

5.32 inc/graphics/widgets/Text.hh File Reference

Allow you to create a text object.

```
#include "type_compatibility.hh"
#include "graphics_header.hh"
#include "GUIButtonId.hh"
#include "CoreManager.hh"
#include "IWidget.hh"
```

Classes

- class Text

  Inherited from a IWidget.

5.32.1 Detailed Description

Allow you to create a text object.

Author

Nicolas-Emmanuel Robert

Version

1.0

5.33 inc/graphics/Window.hh File Reference

Define a window.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include <string>
#include <vector>
#include "type_compatibility.hh"
#include "CoreManager.hh"
```

Classes

- class Window

  Represent the window that will print some scene (view of the application).

5.33.1 Detailed Description

Define a window.

Author

Théophile Champion

Version

1.0

5.34 inc/managers/config/Model.hh File Reference

The Class Model allow you to manipulate model's informations.

```
#include <vector>
#include "type_compatibility.hh"
```

5.34.1 Detailed Description

The Class Model allow you to manipulate model's informations.

The Class Video allow you to manipulate video's configuration.

The Class Sound allow you to manipulate sound's configuration.

The Class Score allow you to manipulate score's informations.

Author

Nicolas-Emmanuel Robert

Version

1.0

5.35 inc/managers/ConfigManager.hh File Reference

Allow you to load, modify and save the configuration.

```
#include <irrlicht/irrlicht.h>
#include <irrlicht/irrXML.h>
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
#include <map>
#include "Control.hh"
#include "Video.hh"
#include "Sound.hh"
#include "Exception.hh"
#include "IControl.hh"
```

Classes

- class ConfigManager

  This class contain all the methode to manipulate the configuration.

5.35.1 Detailed Description

Allow you to load, modify and save the configuration.

Author

Nicolas-Emmanuel Robert

Version

1.0

5.36 inc/managers/CoreManager.hh File Reference

CoreManager regroups all the application's manager.

```
#include "ModelsManager.hh"
#include "ConfigManager.hh"
#include "ScoreManager.hh"
#include "LevelManager.hh"
#include "SaveManager.hh"
#include "EventManager.hh"
```

Classes

- class CoreManager

  CoreManager is the main manager who regroups all the other manager.

5.36.1 Detailed Description

CoreManager regroups all the application's manager.

Author

Théophile Champion

Version

1.0

5.37 inc/managers/LevelManager.hh File Reference

Load level configuration out of XML format.

```
#include <vector>
#include <string>
#include <iostream>
#include <fstream>
#include <irrlicht/irrlicht.h>
#include <irrlicht/irrXML.h>
#include "LevelDefinition.hh"
#include "Exception.hh"
```

5.37.1 Detailed Description

Load level configuration out of XML format.

Author

Charles && Clovis

Version

1.0

5.38 inc/managers/ModelsManager.hh File Reference

The Models manager will allow you to get all the information to load a model.

```
#include <irrlicht/irrlicht.h>
#include <irrlicht/irrXML.h>
#include <iostream>
#include <fstream>
#include <map>
#include "type_compatibility.hh"
#include "Exception.hh"
#include "Model.hh"
```

Classes

- class ModelsManager

  The class to load all the information about a model.

5.38.1 Detailed Description

The Models manager will allow you to get all the information to load a model.

Author

Nicolas-Emmanuel Robert

Version

1.1

5.39 inc/managers/SaveManager.hh File Reference

The save manager will allow you to save and restore a game.

```
#include <memory>
#include "type_compatibility.hh"
```

Classes

- class SaveManager

  The SaveManager class.

5.39.1 Detailed Description

The save manager will allow you to save and restore a game.

Author

Nicolas-Emmanuel Robert

Version

1.0

5.40 inc/managers/ScoreManager.hh File Reference

The score manager will allow you to save, load and restore game scores.

```
#include <vector>
#include <string>
#include <iostream>
#include <fstream>
#include <irrlicht/irrlicht.h>
#include <irrlicht/irrXML.h>
#include "Score.hh"
#include "Exception.hh"
```

Classes

- class ScoreManager

  The ScoreManager class.

5.40.1 Detailed Description

The score manager will allow you to save, load and restore game scores.

Author

Nicolas-Emmanuel Robert

Version

1.0

Index