

Arcade

Generated by Doxygen 1.9.8

1 Class Index	1
1.1 Class List	1
2 File Index	3
2.1 File List	3
3 Class Documentation	5
3.1 Arcade::Core Class Reference	5
3.2 Game1 Class Reference	5
3.3 Arcade::IDisplayModule Class Reference	5
3.3.1 Detailed Description	6
3.3.2 Member Function Documentation	6
3.3.2.1 clear()	6
3.3.2.2 getInput()	6
3.3.2.3 init()	7
3.3.2.4 render()	7
3.3.2.5 setEntities()	7
3.4 Arcade::IGameModule Class Reference	7
3.4.1 Detailed Description	8
3.4.2 Member Function Documentation	8
3.4.2.1 getEntity()	8
3.4.2.2 handleInput()	8
3.4.2.3 render()	8
3.4.2.4 start()	8
3.4.2.5 update()	8
3.5 InputModel Class Reference	9
3.5.1 Detailed Description	9
3.5.2 Member Enumeration Documentation	9
3.5.2.1 Input	9
3.5.3 Member Function Documentation	10
3.5.3.1 getInput()	10
4 File Documentation	11
4.1 core.hpp	11
4.2 DLoader.hpp	11
4.3 GameLoop.hpp	12
4.4 Game1.hpp	12
4.5 Game2.hpp	12
4.6 Game3.hpp	12
4.7 nCurses.hpp	12
4.8 SDL.hpp	13
4.9 SFML.hpp	13
4.10 /home/stesson/Epitech/B-OOP-300/Arcade/src/Shared/Interface/IDisplayModule.hpp File Reference	13

4.10.1 Detailed Description	14
4.11 IDisplayModule.hpp	14
4.12 /home/stesson/Epitech/B-OOP-300/Arcade/src/Shared/Interface/IGameModule.hpp File Reference .	15
4.12.1 Detailed Description	16
4.13 IGameModule.hpp	16
4.14 /home/stesson/Epitech/B-OOP-300/Arcade/src/Shared/Models/InputModel.hpp File Reference . . .	16
4.14.1 Detailed Description	17
4.15 InputModel.hpp	17
Index	19

Chapter 1

Class Index

1.1 Class List

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

Arcade::Core	5
Game1	5
Arcade::IDisplayModule	
Interface for graphical display modules	5
Arcade::IGameModule	
Interface for game logic modules	7
InputModel	
Represents a model for user input handling	9

Chapter 2

File Index

2.1 File List

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

/home/stesson/Epitech/B-OOP-300/Arcade/src/Core/ core.hpp	11
/home/stesson/Epitech/B-OOP-300/Arcade/src/Core/DLoader/ DLoader.hpp	11
/home/stesson/Epitech/B-OOP-300/Arcade/src/Core/GameLoop/ GameLoop.hpp	12
/home/stesson/Epitech/B-OOP-300/Arcade/src/Games/Game1/ Game1.hpp	12
/home/stesson/Epitech/B-OOP-300/Arcade/src/Games/Game2/ Game2.hpp	12
/home/stesson/Epitech/B-OOP-300/Arcade/src/Games/Game3/ Game3.hpp	12
/home/stesson/Epitech/B-OOP-300/Arcade/src/Graphics/nCurses/ nCurses.hpp	12
/home/stesson/Epitech/B-OOP-300/Arcade/src/Graphics/SDL/ SDL.hpp	13
/home/stesson/Epitech/B-OOP-300/Arcade/src/Graphics/SFML/ SFML.hpp	13
/home/stesson/Epitech/B-OOP-300/Arcade/src/Shared/Interface/ IDisplayModule.hpp Interface for display modules in the Arcade project	13
/home/stesson/Epitech/B-OOP-300/Arcade/src/Shared/Interface/ IGameModule.hpp Interface for game modules in the Arcade project	15
/home/stesson/Epitech/B-OOP-300/Arcade/src/Shared/Models/ InputModel.hpp Defines the InputModel class for handling user input in the Arcade project	16

Chapter 3

Class Documentation

3.1 Arcade::Core Class Reference

Public Member Functions

- void **run** ()
- void **loadGame** (const std::string &gameName)
- void **loadDisplay** (const std::string &displayName)
- void **unloadGame** ()
- void **unloadDisplay** ()
- void **printHelp** ()

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

- /home/stesson/Epitech/B-OOP-300/Arcade/src/Core/core.hpp
- /home/stesson/Epitech/B-OOP-300/Arcade/src/Core/core.cpp

3.2 Game1 Class Reference

Public Member Functions

- void **printHelp** ()

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

- /home/stesson/Epitech/B-OOP-300/Arcade/src/Games/Game1/Game1.hpp
- /home/stesson/Epitech/B-OOP-300/Arcade/src/Games/Game1/Game1.cpp

3.3 Arcade::IDisplayModule Class Reference

Interface for graphical display modules.

```
#include <IDisplayModule.hpp>
```

Public Member Functions

- virtual `~IDisplayModule()`=default
Virtual destructor for the interface.
- virtual void `init()`=0
Initializes the display module.
- virtual void `render()`=0
Renders the current frame.
- virtual void `clear()`=0
Clears the display.
- virtual `InputModel` `getInput()`=0
Retrieves the latest user input.
- virtual void `setEntities()`=0
Sets the entities to be displayed.

3.3.1 Detailed Description

Interface for graphical display modules.

The `IDisplayModule` interface provides a set of pure virtual functions that must be implemented by any graphical display module in the Arcade project. It includes methods for initialization, rendering, input handling, and refreshing the display.

3.3.2 Member Function Documentation

3.3.2.1 `clear()`

```
virtual void Arcade::IDisplayModule::clear ( ) [pure virtual]
```

Clears the display.

This method should remove all elements from the screen before the next rendering cycle.

3.3.2.2 `getInput()`

```
virtual InputModel Arcade::IDisplayModule::getInput ( ) [pure virtual]
```

Retrieves the latest user input.

This method captures and returns the latest input received from the user.

Returns

An `InputModel` representing the user's input.

3.3.2.3 init()

```
virtual void Arcade::IDisplayModule::init ( ) [pure virtual]
```

Initializes the display module.

This method is responsible for setting up the display module, loading necessary resources, and preparing the graphical environment.

3.3.2.4 render()

```
virtual void Arcade::IDisplayModule::render ( ) [pure virtual]
```

Renders the current frame.

This method should draw all necessary elements onto the screen.

3.3.2.5 setEntities()

```
virtual void Arcade::IDisplayModule::setEntities ( ) [pure virtual]
```

Sets the entities to be displayed.

This method updates the graphical entities that should be rendered on the screen.

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

- [/home/stesson/Epitech/B-OOP-300/Arcade/src/Shared/Interface/IDisplayModule.hpp](#)

3.4 Arcade::IGameModule Class Reference

Interface for game logic modules.

```
#include <IGameModule.hpp>
```

Public Member Functions

- virtual **~IGameModule** ()=default
Virtual destructor for the interface.
- virtual void **start** ()=0
Starts the game module.
- virtual void **update** ()=0
Updates the game state.
- virtual void **render** ()=0
Renders the game elements.
- virtual void **getEntity** ()=0
Retrieves the game entities.
- virtual void **handleInput** ()=0
Handles player input.

3.4.1 Detailed Description

Interface for game logic modules.

The [IGameModule](#) interface provides a set of pure virtual functions that must be implemented by any game module in the Arcade project. It includes methods for starting, updating, rendering, and handling input.

3.4.2 Member Function Documentation

3.4.2.1 `getEntity()`

```
virtual void Arcade::IGameModule::getEntity ( ) [pure virtual]
```

Retrieves the game entities.

This method should return the current entities involved in the game to be processed or displayed.

3.4.2.2 `handleInput()`

```
virtual void Arcade::IGameModule::handleInput ( ) [pure virtual]
```

Handles player input.

This method should process user input and apply the corresponding actions to the game logic.

3.4.2.3 `render()`

```
virtual void Arcade::IGameModule::render ( ) [pure virtual]
```

Renders the game elements.

This method should be responsible for preparing the game's visual elements before they are displayed.

3.4.2.4 `start()`

```
virtual void Arcade::IGameModule::start ( ) [pure virtual]
```

Starts the game module.

This method is responsible for initializing the game state and preparing all necessary resources for the game session.

3.4.2.5 `update()`

```
virtual void Arcade::IGameModule::update ( ) [pure virtual]
```

Updates the game state.

This method should process game logic, update entities, and handle internal state transitions.

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

- [/home/stesson/Epitech/B-OOP-300/Arcade/src/Shared/Interface/IGameModule.hpp](#)

3.5 InputModel Class Reference

Represents a model for user input handling.

```
#include <InputModel.hpp>
```

Public Types

- enum [Input](#) {
 UP, DOWN, LEFT, RIGHT,
 ENTER, ESCAPE, SPACE, A,
 B, C, D, E,
 F, G, H, I,
 J, K, L, M,
 N, O, P, Q,
 R, S, T, U,
 V, W, X, Y,
 Z, ZERO, ONE, TWO,
 THREE, FOUR, FIVE, SIX,
 SEVEN, EIGHT, NINE, UNKNOWN,
 RIGHT_CLICK, LEFT_CLICK, MIDDLE_CLICK }

Defines various user input types.

Public Member Functions

- [Input](#) getInput () const
Retrieves the current user input.

3.5.1 Detailed Description

Represents a model for user input handling.

The [InputModel](#) class defines an enumeration of possible inputs and provides a method to retrieve the current input.

3.5.2 Member Enumeration Documentation

3.5.2.1 Input

```
enum InputModel::Input
```

Defines various user input types.

This enumeration includes keyboard keys, mouse buttons, and an UNKNOWN value for unrecognized inputs.

Enumerator

RIGHT	Directional movement keys.
SPACE	Special action keys.
Z	Alphabetic keys.
NINE	Numeric keys.
UNKNOWN	Unrecognized input.
MIDDLE_CLICK	Mouse buttons.

3.5.3 Member Function Documentation

3.5.3.1 getInput()

`Input` InputModel::getInput () const

Retrieves the current user input.

This method returns the latest detected input from the user.

Returns

An Input enumeration value representing the current input.

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

- [/home/stesson/Epitech/B-OOP-300/Arcade/src/Shared/Models/InputModel.hpp](#)

Chapter 4

File Documentation

4.1 core.hpp

```
00001 // Copyright 2025 <Epitech>
00002 /*
00003  ** EPITECH PROJECT, 2025
00004  ** Arcade
00005  ** File description:
00006  ** core
00007  */
00008
00009 #ifndef SRC_CORE_CORE_HPP_
00010     #define SRC_CORE_CORE_HPP_
00011     #include <memory>
00012     #include <string>
00013     #include "../Shared/Interface/IGameModule.hpp"
00014     #include "../Shared/Interface/IDisplayModule.hpp"
00015
00016 namespace Arcade {
00017     class Core {
00018     public:
00019         Core() : _currentGame(nullptr), _currentDisplay(nullptr) {}
00020         ~Core() {}
00021         void run();
00022         void loadGame(const std::string &gameName);
00023         void loadDisplay(const std::string &displayName);
00024         void unloadGame();
00025         void unloadDisplay();
00026         void printHelp();
00027
00028     private:
00029         std::unique_ptr<IGameModule> _currentGame;
00030         std::unique_ptr<IDisplayModule> _currentDisplay;
00031     };
00032 } // namespace Arcade
00033 #endif // SRC_CORE_CORE_HPP_
```

4.2 DLoader.hpp

```
00001 // Copyright 2025 <Epitech>
00002 /*
00003  ** EPITECH PROJECT, 2025
00004  ** B-OOP-400 Arcade
00005  ** File description:
00006  ** DLoader class
00007  */
00008
00009 #ifndef SRC_CORE_DLOADER_DLOADER_HPP_
00010     #define SRC_CORE_DLOADER_DLOADER_HPP_
00011
00012 #endif // SRC_CORE_DLOADER_DLOADER_HPP_
```

4.3 GameLoop.hpp

```
00001 // Copyright 2025 <Epitech>
00002 /*
00003 ** EPITECH PROJECT, 2025
00004 ** B-OOP-400 Arcade
00005 ** File description:
00006 ** GameLoop class
00007 */
00008
00009 #ifndef SRC_CORE_GAMELOOP_GAMELOOP_HPP_
00010     #define SRC_CORE_GAMELOOP_GAMELOOP_HPP_
00011
00012 #endif // SRC_CORE_GAMELOOP_GAMELOOP_HPP_
```

4.4 Game1.hpp

```
00001 // Copyright 2025 <Epitech>
00002 /*
00003 ** EPITECH PROJECT, 2025
00004 ** B-OOP-400 Arcade
00005 ** File description:
00006 ** Game1 class
00007 */
00008
00009 #ifndef SRC_GAMES_GAME1_GAME1_HPP_
00010     #define SRC_GAMES_GAME1_GAME1_HPP_
00011
00012     class Game1 {
00013     public:
00014         Game1() = default;
00015         ~Game1() = default;
00016         void printHelp();
00017     };
00018
00019 #endif // SRC_GAMES_GAME1_GAME1_HPP_
```

4.5 Game2.hpp

```
00001 // Copyright 2025 <Epitech>
00002 /*
00003 ** EPITECH PROJECT, 2025
00004 ** B-OOP-400 Arcade
00005 ** File description:
00006 ** Game2
00007 */
00008
00009 #ifndef SRC_GAMES_GAME2_GAME2_HPP_
00010     #define SRC_GAMES_GAME2_GAME2_HPP_
00011
00012 #endif // SRC_GAMES_GAME2_GAME2_HPP_
```

4.6 Game3.hpp

```
00001 // Copyright 2025 <Epitech>
00002 /*
00003 ** EPITECH PROJECT, 2025
00004 ** B-OOP-400 Arcade
00005 ** File description:
00006 ** Game3 class
00007 */
00008 #ifndef SRC_GAMES_GAME3_GAME3_HPP_
00009     #define SRC_GAMES_GAME3_GAME3_HPP_
00010
00011 #endif // SRC_GAMES_GAME3_GAME3_HPP_
```

4.7 nCurses.hpp

```
00001 // Copyright 2025 <Epitech>
00002 /*
```



```

00003 ** EPITECH PROJECT, 2025
00004 ** B-OOP-400 Arcade
00005 ** File description:
00006 ** nCurses class
00007 */
00008
00009 #ifndef SRC_GRAPHICS_NCURRESES_NCURRESES_HPP_
00010     #define SRC_GRAPHICS_NCURRESES_NCURRESES_HPP_
00011
00012 #endif // SRC_GRAPHICS_NCURRESES_NCURRESES_HPP_

```

4.8 SDL.hpp

```

00001 // Copyright 2025 <Epitech>
00002 /*
00003 ** EPITECH PROJECT, 2025
00004 ** B-OOP-400 Arcade
00005 ** File description:
00006 ** SDL class
00007 */
00008
00009 #ifndef SRC_GRAPHICS_SDL_SDL_HPP_
00010     #define SRC_GRAPHICS_SDL_SDL_HPP_
00011
00012 #endif // SRC_GRAPHICS_SDL_SDL_HPP_

```

4.9 SFML.hpp

```

00001 // Copyright 2025 <Epitech>
00002 /*
00003 ** EPITECH PROJECT, 2025
00004 ** B-OOP-400 Arcade
00005 ** File description:
00006 ** SFML class
00007 */
00008
00009 #ifndef SRC_GRAPHICS_SFML_SFML_HPP_
00010     #define SRC_GRAPHICS_SFML_SFML_HPP_
00011
00012 #endif // SRC_GRAPHICS_SFML_SFML_HPP_

```

4.10 /home/stesson/Epitech/B-OOP-300/Arcade/src/Shared/Interface/IDisplayModule.hpp File Reference

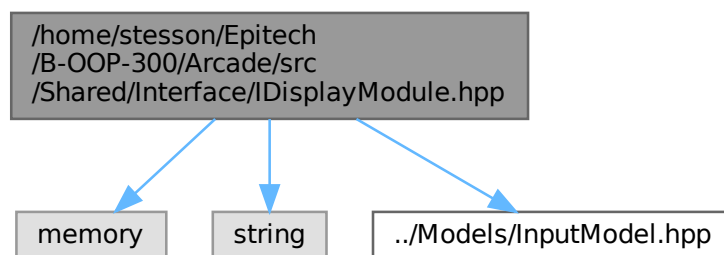
Interface for display modules in the Arcade project.

```

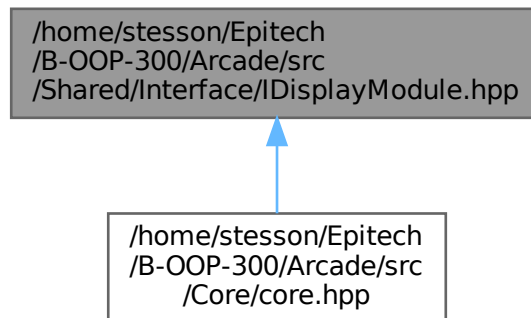
#include <memory>
#include <string>
#include "../Models/InputModel.hpp"

```

Include dependency graph for IDisplayModule.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class [Arcade::IDisplayModule](#)
Interface for graphical display modules.

4.10.1 Detailed Description

Interface for display modules in the Arcade project.

This interface defines the necessary methods for implementing a display module in the Arcade project.

4.11 IDisplayModule.hpp

[Go to the documentation of this file.](#)

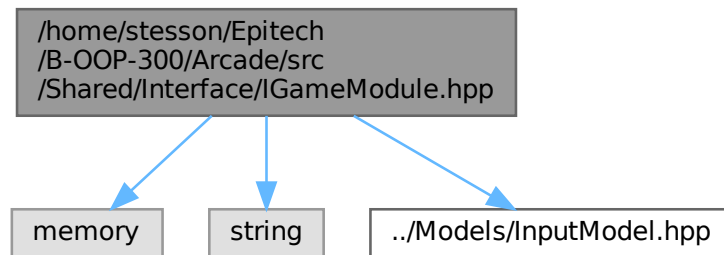
```

00001 // Copyright 2025 <Epitech>
00002 /*
00003  ** EPITECH PROJECT, 2025
00004  ** B-OOP-400 Arcade
00005  ** File description:
00006  ** IDisplayModule interface
00007  */
00015 #ifndef SRC_SHARED_INTERFACE_IDISPLAYMODULE_HPP_
00016     #define SRC_SHARED_INTERFACE_IDISPLAYMODULE_HPP_
00017     #include <memory>
00018     #include <string>
00019     #include "../Models/InputModel.hpp"
00020
00021 namespace Arcade {
00031     class IDisplayModule {
00032     public:
00036         virtual ~IDisplayModule() = default;
00043         virtual void init() = 0;
00049         virtual void render() = 0;
00056         virtual void clear() = 0;
00063         virtual InputModel getInput() = 0;
00069         virtual void setEntities() = 0;
00070     };
00071 } // namespace Arcade
00072 #endif // SRC_SHARED_INTERFACE_IDISPLAYMODULE_HPP_
  
```

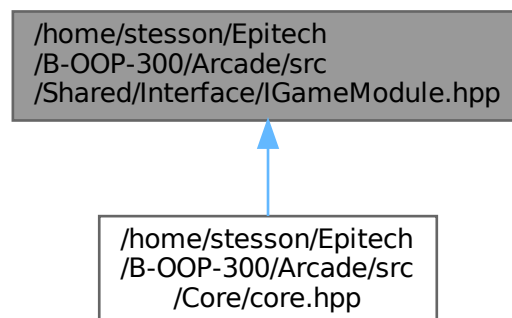
4.12 /home/stesson/Epitech/B-OOP-300/Arcade/src/Shared/Interface/IGameModule.hpp File Reference

Interface for game modules in the Arcade project.

```
#include <memory>
#include <string>
#include "../Models/InputModel.hpp"
Include dependency graph for IGameModule.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [Arcade::IGameModule](#)
Interface for game logic modules.

4.12.1 Detailed Description

Interface for game modules in the Arcade project.

This interface defines the necessary methods for implementing a game module in the Arcade project.

4.13 IGameModule.hpp

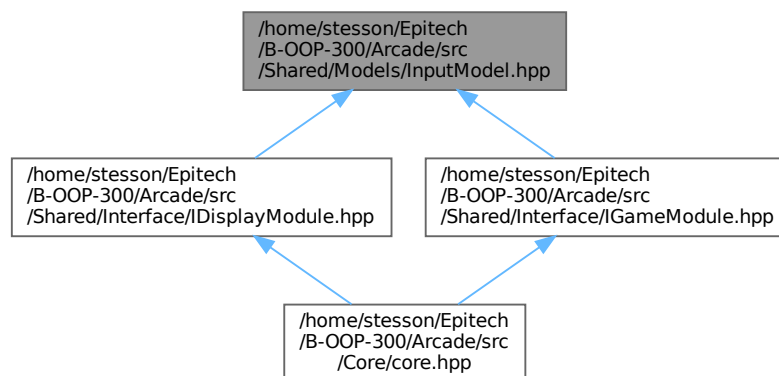
[Go to the documentation of this file.](#)

```
00001 // Copyright 2025 <Epitech>
00002 /*
00003 ** EPITECH PROJECT, 2025
00004 ** B-OOP-400 Arcade
00005 ** File description:
00006 ** IGameModule interface
00007 */
00016 #ifndef SRC_SHARED_INTERFACE_IGAMEMODULE_HPP_
00017 #define SRC_SHARED_INTERFACE_IGAMEMODULE_HPP_
00018     #include <memory>
00019     #include <string>
00020     #include "../Models/InputModel.hpp"
00021
00022 namespace Arcade {
00031 class IGameModule {
00032 public:
00036     virtual ~IGameModule() = default;
00043     virtual void start() = 0;
00044
00051     virtual void update() = 0;
00052
00059     virtual void render() = 0;
00060
00067     virtual void getEntity() = 0;
00068
00075     virtual void handleInput() = 0;
00076 };
00077 } // namespace Arcade
00078 #endif // SRC_SHARED_INTERFACE_IGAMEMODULE_HPP_
```

4.14 /home/stesson/Epitech/B-OOP-300/Arcade/src/Shared/Models/↵ InputModel.hpp File Reference

Defines the [InputModel](#) class for handling user input in the Arcade project.

This graph shows which files directly or indirectly include this file:



Classes

- class [InputModel](#)

Represents a model for user input handling.

4.14.1 Detailed Description

Defines the [InputModel](#) class for handling user input in the Arcade project.

This class provides an enumeration of possible user inputs and a method to retrieve them.

4.15 InputModel.hpp

[Go to the documentation of this file.](#)

```
00001 // Copyright 2025 <Epitech>
00002 /*
00003  ** EPITECH PROJECT, 2025
00004  ** B-OOP-400
00005  ** File description:
00006  ** Input Model
00007  */
00014 #ifndef SRC_SHARED_MODELS_INPUTMODEL_HPP_
00015     #define SRC_SHARED_MODELS_INPUTMODEL_HPP_
00016
00024 class InputModel {
00025     public:
00033         enum Input {
00034             UP, DOWN, LEFT, RIGHT,
00035             ENTER, ESCAPE, SPACE,
00036             A, B, C, D, E, F, G, H, I, J, K, L, M, N, O,
00037             P, Q, R, S, T, U, V, W, X, Y, Z,
00038             ZERO, ONE, TWO, THREE, FOUR, FIVE,
00039             SIX, SEVEN, EIGHT, NINE,
00040             UNKNOWN,
00041             RIGHT_CLICK, LEFT_CLICK, MIDDLE_CLICK
00042         };
00049         Input getInput() const;
00050     };
00051
00052 #endif // SRC_SHARED_MODELS_INPUTMODEL_HPP_
```


Index

/home/stesson/Epitech/B-OOP-300/Arcade/src/Core/DLoader/DLoader.hpp, 8
11
/home/stesson/Epitech/B-OOP-300/Arcade/src/Core/GameLoop/GameLoop.hpp, 11
12
Arcade::IDisplayModule, 6
/home/stesson/Epitech/B-OOP-300/Arcade/src/Core/core.hpp, 11
InputModel, 9
/home/stesson/Epitech/B-OOP-300/Arcade/src/Games/Game1/Game1.hpp, 12
InputModel, 9
getInput, 10
/home/stesson/Epitech/B-OOP-300/Arcade/src/Games/Game2/Game2.hpp, 12
LEFT, 9
MIDDLE_CLICK, 9
/home/stesson/Epitech/B-OOP-300/Arcade/src/Games/Game3/Game3.hpp, 12
NINE, 9
RIGHT, 9
/home/stesson/Epitech/B-OOP-300/Arcade/src/Graphics/SDL/SDL.hpp, 13
SPACE, 9
UNKNOWN, 9
/home/stesson/Epitech/B-OOP-300/Arcade/src/Graphics/SFML/SFML.hpp, 13
Z, 9
/home/stesson/Epitech/B-OOP-300/Arcade/src/Graphics/nCurses/nCurses.hpp, 12
MIDDLE_CLICK, 9
InputModel, 9
/home/stesson/Epitech/B-OOP-300/Arcade/src/Shared/Interface/IDisplayModule.hpp, 13, 14
NINE
InputModel, 9
/home/stesson/Epitech/B-OOP-300/Arcade/src/Shared/Interface/IGameModule.hpp, 15, 16
render
/home/stesson/Epitech/B-OOP-300/Arcade/src/Shared/Models/InputModel.hpp, 16, 17
Arcade::IDisplayModule, 7
Arcade::IGameModule, 8
RIGHT
InputModel, 9
setEntities
Arcade::IDisplayModule, 7
SPACE
InputModel, 9
start
Arcade::IGameModule, 8
UNKNOWN
InputModel, 9
update
Arcade::IGameModule, 8
Z
InputModel, 9
Arcade::Core, 5
Arcade::IDisplayModule, 5
clear, 6
getInput, 6
init, 6
render, 7
setEntities, 7
Arcade::IGameModule, 7
getEntity, 8
handleInput, 8
render, 8
start, 8
update, 8
clear
Arcade::IDisplayModule, 6
Game1, 5
getEntity
Arcade::IGameModule, 8
getInput
Arcade::IDisplayModule, 6
InputModel, 10
handleInput