

R-Type architecture

Generated by Doxygen 1.13.2

| | |
|--|-----------|
| 1 Hierarchical Index | 1 |
| 1.1 Class Hierarchy | 1 |
| 2 Class Index | 3 |
| 2.1 Class List | 3 |
| 3 File Index | 5 |
| 3.1 File List | 5 |
| 4 Class Documentation | 7 |
| 4.1 gsm::AGameState Class Reference | 7 |
| 4.1.1 Member Function Documentation | 7 |
| 4.1.1.1 enter() [1/2] | 7 |
| 4.1.1.2 enter() [2/2] | 8 |
| 4.1.1.3 exit() [1/2] | 8 |
| 4.1.1.4 exit() [2/2] | 8 |
| 4.1.1.5 render() | 8 |
| 4.1.1.6 update() [1/2] | 8 |
| 4.1.1.7 update() [2/2] | 8 |
| 4.2 gsm::AGameStateMachine Class Reference | 9 |
| 4.2.1 Member Function Documentation | 9 |
| 4.2.1.1 changeState() [1/2] | 9 |
| 4.2.1.2 changeState() [2/2] | 9 |
| 4.2.1.3 popState() [1/2] | 9 |
| 4.2.1.4 popState() [2/2] | 10 |
| 4.2.1.5 pushState() [1/2] | 10 |
| 4.2.1.6 pushState() [2/2] | 10 |
| 4.2.1.7 render() | 10 |
| 4.2.1.8 update() [1/2] | 10 |
| 4.2.1.9 update() [2/2] | 10 |
| 4.3 gfx::color_t Struct Reference | 11 |
| 4.4 gfx::IEvent Class Reference | 11 |
| 4.5 gsm::IGameState Class Reference | 11 |
| 4.6 gsm::IGameStateMachine Class Reference | 12 |
| 4.7 net::IServer Class Reference | 12 |
| 4.8 gfx::IWindow Class Reference | 13 |
| 4.9 net::Server Class Reference | 13 |
| 4.9.1 Member Function Documentation | 14 |
| 4.9.1.1 start() | 14 |
| 4.9.1.2 stop() | 14 |
| 4.10 net::ServerConfig Class Reference | 14 |
| 5 File Documentation | 15 |
| 5.1 IEvent.hpp | 15 |

| | |
|-------------------------------------|-----------|
| 5.2 IWindow.hpp | 15 |
| 5.3 AGameStateMachine.hpp | 16 |
| 5.4 AGameStateMachine.hpp | 16 |
| 5.5 IGameStateMachine.hpp | 17 |
| 5.6 IGameStateMachine.hpp | 17 |
| 5.7 AGameState.hpp | 17 |
| 5.8 AGameState.hpp | 18 |
| 5.9 IGameState.hpp | 18 |
| 5.10 IGameState.hpp | 18 |
| 5.11 IServer.hpp | 19 |
| 5.12 Server.hpp | 19 |
| 5.13 ServerConfig.hpp | 19 |
| Index | 21 |

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

| | |
|----------------------------------|----|
| gfx::color_t | 11 |
| gfx::IEvent | 11 |
| gsm::IGameState | 11 |
| gsm::AGameState | 7 |
| gsm::AGameState | 7 |
| gsm::IGameStateMachine | 12 |
| gsm::AGameStateMachine | 9 |
| gsm::AGameStateMachine | 9 |
| net::IServer | 12 |
| net::Server | 13 |
| gfx::IWindow | 13 |
| net::ServerConfig | 14 |

Chapter 2

Class Index

2.1 Class List

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

| | |
|--|----|
| gsm::AGameState | 7 |
| gsm::AGameStateMachine | 9 |
| gfx::color_t | 11 |
| gfx::IEvent | 11 |
| gsm::IGameState | 11 |
| gsm::IGameStateMachine | 12 |
| net::IServer | 12 |
| gfx::IWindow | 13 |
| net::Server | 13 |
| net::ServerConfig | 14 |

Chapter 3

File Index

3.1 File List

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

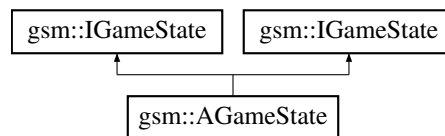
| | |
|--|----|
| /home/albane/epitech/tech3/r-type/ryanR-type/client/graphics/ IEvent.hpp | 15 |
| /home/albane/epitech/tech3/r-type/ryanR-type/client/graphics/ IWindow.hpp | 15 |
| /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/machine/ AGameStateMachine.hpp | 16 |
| /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/machine/ IGameStateMachine.hpp | 17 |
| /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/ AGameState.hpp | 17 |
| /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/ IGameState.hpp | 18 |
| /home/albane/epitech/tech3/r-type/ryanR-type/server/ IServer.hpp | 19 |
| /home/albane/epitech/tech3/r-type/ryanR-type/server/ Server.hpp | 19 |
| /home/albane/epitech/tech3/r-type/ryanR-type/server/ ServerConfig.hpp | 19 |
| /home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/machine/ AGameStateMachine.hpp | 16 |
| /home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/machine/ IGameStateMachine.hpp | 17 |
| /home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/states/ AGameState.hpp | 18 |
| /home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/states/ IGameState.hpp | 18 |

Chapter 4

Class Documentation

4.1 gsm::AGameState Class Reference

Inheritance diagram for gsm::AGameState:



Public Member Functions

- **AGameState** (std::shared_ptr< [IGameStateMachine](#) > gsm)
- void [enter](#) () override
- void [update](#) (float deltaTime) override
- void [render](#) () override
- void [exit](#) () override
- **AGameState** (std::shared_ptr< [IGameStateMachine](#) > gsm)
- void [enter](#) () override
- void [update](#) (float deltaTime) override
- void [exit](#) () override

Protected Attributes

- std::shared_ptr< [IGameStateMachine](#) > **_gsm**

4.1.1 Member Function Documentation

4.1.1.1 enter() [1/2]

```
void gsm::AGameState::enter () [override], [virtual]
```

Implements [gsm::IGameState](#).

4.1.1.2 enter() [2/2]

```
void gsm::AGameState::enter () [override], [virtual]
```

Implements [gsm::IGameState](#).

4.1.1.3 exit() [1/2]

```
void gsm::AGameState::exit () [override], [virtual]
```

Implements [gsm::IGameState](#).

4.1.1.4 exit() [2/2]

```
void gsm::AGameState::exit () [override], [virtual]
```

Implements [gsm::IGameState](#).

4.1.1.5 render()

```
void gsm::AGameState::render () [override], [virtual]
```

Implements [gsm::IGameState](#).

4.1.1.6 update() [1/2]

```
void gsm::AGameState::update (  
    float deltaTime) [override], [virtual]
```

Implements [gsm::IGameState](#).

4.1.1.7 update() [2/2]

```
void gsm::AGameState::update (  
    float deltaTime) [override], [virtual]
```

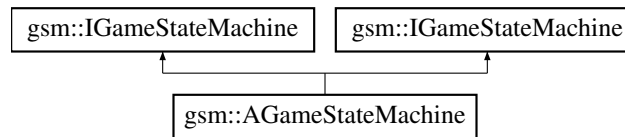
Implements [gsm::IGameState](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/AGameState.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/states/AGameState.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/AGameState.cpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/states/AGameState.cpp

4.2 gsm::AGameStateMachine Class Reference

Inheritance diagram for gsm::AGameStateMachine:



Public Member Functions

- void [changeState](#) (std::shared_ptr< [IGameState](#) > newState) override
- void [pushState](#) (std::shared_ptr< [IGameState](#) > newState) override
- void [popState](#) () override
- void [update](#) (float deltaTime) override
- void [render](#) () override
- void [changeState](#) (std::shared_ptr< [IGameState](#) > newState) override
- void [pushState](#) (std::shared_ptr< [IGameState](#) > newState) override
- void [popState](#) () override
- void [update](#) (float deltaTime) override

Protected Attributes

- std::stack< std::shared_ptr< [IGameState](#) > > **_states**

4.2.1 Member Function Documentation

4.2.1.1 changeState() [1/2]

```
void gsm::AGameStateMachine::changeState (
    std::shared_ptr< IGameState > newState) [override], [virtual]
```

Implements [gsm::IGameStateMachine](#).

4.2.1.2 changeState() [2/2]

```
void gsm::AGameStateMachine::changeState (
    std::shared_ptr< IGameState > newState) [override], [virtual]
```

Implements [gsm::IGameStateMachine](#).

4.2.1.3 popState() [1/2]

```
void gsm::AGameStateMachine::popState () [override], [virtual]
```

Implements [gsm::IGameStateMachine](#).

4.2.1.4 popState() [2/2]

```
void gsm::AGameStateMachine::popState () [override], [virtual]
```

Implements [gsm::IGameStateMachine](#).

4.2.1.5 pushState() [1/2]

```
void gsm::AGameStateMachine::pushState (  
    std::shared_ptr< IGameState > newState) [override], [virtual]
```

Implements [gsm::IGameStateMachine](#).

4.2.1.6 pushState() [2/2]

```
void gsm::AGameStateMachine::pushState (  
    std::shared_ptr< IGameState > newState) [override], [virtual]
```

Implements [gsm::IGameStateMachine](#).

4.2.1.7 render()

```
void gsm::AGameStateMachine::render () [override], [virtual]
```

Implements [gsm::IGameStateMachine](#).

4.2.1.8 update() [1/2]

```
void gsm::AGameStateMachine::update (  
    float deltaTime) [override], [virtual]
```

Implements [gsm::IGameStateMachine](#).

4.2.1.9 update() [2/2]

```
void gsm::AGameStateMachine::update (  
    float deltaTime) [override], [virtual]
```

Implements [gsm::IGameStateMachine](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/machine/AGameStateMachine.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/machine/AGameStateMachine.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/machine/AGameStateMachine.cpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/machine/AGameStateMachine.cpp

4.3 gfx::color_t Struct Reference

Public Attributes

- int **r**
- int **g**
- int **b**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/graphics/IWindow.hpp

4.4 gfx::IEvent Class Reference

Public Types

- enum **event_e** {
UP , **DOWN** , **LEFT** , **RIGHT** ,
SPACE , **ENTER** , **ESCAPE** , **CLOSE** ,
NEXTGAME , **NEXTGRAPHIC** , **REFRESH** , **MOUSECLICK** ,
MOUSERIGHTCLICK , **MOUSELEFTCLICK** , **MENU** , **NOTHING** ,
TYIPING , **NEXTDIFFICULTY** }
 • typedef enum gfx::IEvent::event_e **event_t**

Public Member Functions

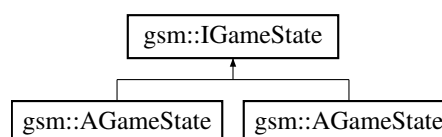
- virtual void **init** ()=0
- virtual event_t **pollEvents** (std::pair< int, int > gridSize)=0
- virtual void **cleanup** ()=0
- virtual std::pair< int, int > **getMousePos** ()=0
- virtual void **setMapSize** (std::pair< int, int > size)=0

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/graphics/IEvent.hpp

4.5 gsm::IGameState Class Reference

Inheritance diagram for gsm::IGameState:



Public Member Functions

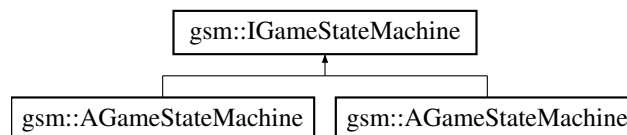
- virtual void **enter** ()=0
- virtual void **update** (float deltaTime)=0
- virtual void **render** ()=0
- virtual void **exit** ()=0
- virtual void **enter** ()=0
- virtual void **update** (float deltaTime)=0
- virtual void **exit** ()=0

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/IGameState.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/states/IGameState.hpp

4.6 gsm::IGameStateMachine Class Reference

Inheritance diagram for gsm::IGameStateMachine:



Public Member Functions

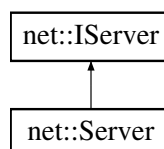
- virtual void **changeState** (std::shared_ptr< [IGameState](#) > newState)=0
- virtual void **pushState** (std::shared_ptr< [IGameState](#) > newState)=0
- virtual void **popState** ()=0
- virtual void **update** (float deltaTime)=0
- virtual void **render** ()=0
- virtual void **changeState** (std::shared_ptr< [IGameState](#) > newState)=0
- virtual void **pushState** (std::shared_ptr< [IGameState](#) > newState)=0
- virtual void **popState** ()=0
- virtual void **update** (float deltaTime)=0

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/machine/IGameStateMachine.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/machine/IGameStateMachine.hpp

4.7 net::IServer Class Reference

Inheritance diagram for net::IServer:



Public Member Functions

- virtual void **start** ()=0
- virtual void **stop** ()=0

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

- /home/albane/epitech/tech3/r-type/ryanR-type/server/IServer.hpp

4.8 gfx::IWindow Class Reference

Public Member Functions

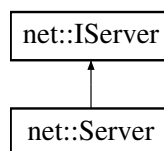
- virtual void **display** ()=0
- virtual void **closeWindow** ()=0
- virtual bool **isOpen** ()=0
- virtual void **clear** ()=0
- virtual void **setMapSize** (std::pair< size_t, size_t > size)=0
- virtual void **resizeWindow** (size_t x, size_t y)=0
- virtual void **drawSprite** (std::string asset, int color, std::string text, std::pair< size_t, size_t > position)=0
- virtual void **drawText** (std::string text, int color, std::pair< size_t, size_t > position)=0
- virtual void **drawRectangle** (int color, std::pair< size_t, size_t > position)=0
- virtual bool **isMouseOver** (std::pair< size_t, size_t > position, std::pair< size_t, size_t > size)=0
- virtual std::pair< int, int > **getWindowSize** ()=0

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/graphics/IWindow.hpp

4.9 net::Server Class Reference

Inheritance diagram for net::Server:

**Public Member Functions**

- void **start** () override
- void **stop** () override

4.9.1 Member Function Documentation

4.9.1.1 start()

```
void net::Server::start () [override], [virtual]
```

Implements [net::IServer](#).

4.9.1.2 stop()

```
void net::Server::stop () [override], [virtual]
```

Implements [net::IServer](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/server/Server.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/Server.cpp

4.10 net::ServerConfig Class Reference

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

- /home/albane/epitech/tech3/r-type/ryanR-type/server/ServerConfig.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/ServerConfig.cpp

Chapter 5

File Documentation

5.1 IEvent.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** r-type
00004 ** File description:
00005 ** IEvent
00006 */
00007
00008 #ifndef IEVENT_HPP_
00009 #define IEVENT_HPP_
00010
00011 #include <utility>
00012 #include <memory>
00013
00014 namespace gfx {
00015
00016 class IEvent {
00017     public:
00018         typedef enum event_e {
00019             UP,
00020             DOWN,
00021             LEFT,
00022             RIGHT,
00023             SPACE,
00024             ENTER,
00025             ESCAPE,
00026             CLOSE,
00027             NEXTGAME,
00028             NEXTGRAPHIC,
00029             REFRESH,
00030             MOUSECLICK,
00031             MOUSERIGHTCLICK,
00032             MOUSELEFTCLICK,
00033             MENU,
00034             NOTHING,
00035             TYIPING,
00036             NEXTDIFFICULTY
00037         } event_t;
00038         virtual ~IEvent() = default;
00039         virtual void init() = 0;
00040         virtual event_t pollEvents(std::pair<int, int> gridSize) = 0;
00041         virtual void cleanup() = 0;
00042         virtual std::pair<int, int> getMousePos() = 0;
00043         virtual void setMapSize(std::pair<int, int> size) = 0;
00044
00045 };
00046
00047 } // namespace gfx
00048
00049 #endif /* !IEVENT_HPP_ */
```

5.2 IWindow.hpp

```
00001 /*
```

```

00002 ** EPITECH PROJECT, 2025
00003 ** r-type
00004 ** File description:
00005 ** IWindow
00006 */
00007
00008
00009 #ifndef IWINDO_HPP_
00010 #define IWINDO_HPP_
00011
00012 #include <string>
00013 #include <utility>
00014 #include <memory>
00015
00016 namespace gfx {
00017
00018 struct color_t {
00019     int r;
00020     int g;
00021     int b;
00022 };
00023
00024 class IWindow {
00025     public:
00026         virtual ~IWindow() = default;
00027         virtual void display() = 0;
00028         virtual void closeWindow() = 0;
00029         virtual bool isOpen() = 0;
00030         virtual void clear() = 0;
00031
00032         virtual void setMapSize(std::pair<size_t, size_t> size) = 0;
00033         virtual void resizeWindow(size_t x, size_t y) = 0;
00034
00035         virtual void drawSprite(std::string asset, int color, std::string text, std::pair<size_t,
00036 size_t> position) = 0;
00037         virtual void drawText(std::string text, int color, std::pair<size_t, size_t> position) = 0;
00038         virtual void drawRectangle(int color, std::pair<size_t, size_t> position) = 0;
00039
00040         virtual bool isMouseOver(std::pair<size_t, size_t> position, std::pair<size_t, size_t> size) =
0;
00041         virtual std::pair<int, int> getWindowSize() = 0;
00042 };
00043 } // namespace gfx
00044
00045 #endif /* !IWINDO_HPP_ */

```

5.3 AGameStateMachine.hpp

```

00001 #pragma once
00002
00003 #include "IGameStateMachine.hpp"
00004 #include "../states/IGameState.hpp"
00005
00006 namespace gsm {
00007
00008 class AGameStateMachine : public IGameStateMachine {
00009     public:
00010         AGameStateMachine();
00011         ~AGameStateMachine() override = default;
00012
00013         void changeState(std::shared_ptr<IGameState> newState) override;
00014         void pushState(std::shared_ptr<IGameState> newState) override;
00015         void popState() override;
00016
00017         void update(float deltaTime) override;
00018         void render() override;
00019
00020     protected:
00021         std::stack<std::shared_ptr<IGameState>> _states;
00022 };
00023
00024 } // namespace gsm

```

5.4 AGameStateMachine.hpp

```

00001 #pragma once
00002

```

```

00003 #include "IGameStateMachine.hpp"
00004 #include "../states/IGameState.hpp"
00005
00006 namespace gsm {
00007
00008 class AGameStateMachine : public IGameStateMachine {
00009 public:
00010     AGameStateMachine();
00011     ~AGameStateMachine() override = default;
00012
00013     void changeState(std::shared_ptr<IGameState> newState) override;
00014     void pushState(std::shared_ptr<IGameState> newState) override;
00015     void popState() override;
00016
00017     void update(float deltaTime) override;
00018
00019 protected:
00020     std::stack<std::shared_ptr<IGameState> _states;
00021 };
00022
00023 } // namespace gsm

```

5.5 IGameStateMachine.hpp

```

00001 #pragma once
00002
00003 #include <memory>
00004 #include <stack>
00005
00006 namespace gsm {
00007
00008 class IGameState;
00009
00010 class IGameStateMachine {
00011 public:
00012     virtual ~IGameStateMachine() = default;
00013
00014     virtual void changeState(std::shared_ptr<IGameState> newState) = 0;
00015     virtual void pushState(std::shared_ptr<IGameState> newState) = 0;
00016     virtual void popState() = 0;
00017
00018     virtual void update(float deltaTime) = 0;
00019     virtual void render() = 0;
00020 };
00021
00022 } // namespace gsm

```

5.6 IGameStateMachine.hpp

```

00001 #pragma once
00002
00003 #include <memory>
00004 #include <stack>
00005
00006 namespace gsm {
00007
00008 class IGameState;
00009
00010 class IGameStateMachine {
00011 public:
00012     virtual ~IGameStateMachine() = default;
00013
00014     virtual void changeState(std::shared_ptr<IGameState> newState) = 0;
00015     virtual void pushState(std::shared_ptr<IGameState> newState) = 0;
00016     virtual void popState() = 0;
00017
00018     virtual void update(float deltaTime) = 0;
00019 };
00020
00021 } // namespace gsm

```

5.7 AGameState.hpp

```

00001 #pragma once

```

```

00002
00003 #include "IGameState.hpp"
00004
00005 namespace gsm {
00006
00007 class AGameState : public IGameState {
00008 public:
00009     AGameState(std::shared_ptr<IGameStateMachine> gsm);
00010     ~AGameState() override = default;
00011
00012     void enter() override;
00013     void update(float deltaTime) override;
00014     void render() override;
00015     void exit() override;
00016
00017 protected:
00018     std::shared_ptr<IGameStateMachine> _gsm;
00019 };
00020
00021 } // namespace gsm

```

5.8 AGameState.hpp

```

00001 #pragma once
00002
00003 #include "IGameState.hpp"
00004
00005 namespace gsm {
00006
00007 class AGameState : public IGameState {
00008 public:
00009     AGameState(std::shared_ptr<IGameStateMachine> gsm);
00010     ~AGameState() override = default;
00011
00012     void enter() override;
00013     void update(float deltaTime) override;
00014     void exit() override;
00015
00016 protected:
00017     std::shared_ptr<IGameStateMachine> _gsm;
00018 };
00019
00020 } // namespace gsm

```

5.9 IGameState.hpp

```

00001 #pragma once
00002
00003 #include <memory>
00004 #include "../machine/IGameStateMachine.hpp"
00005
00006 namespace gsm {
00007
00008 class IGameState {
00009 public:
00010     virtual ~IGameState() = default;
00011
00012     virtual void enter() = 0;
00013     virtual void update(float deltaTime) = 0;
00014     virtual void render() = 0;
00015     virtual void exit() = 0;
00016 };
00017
00018 } // namespace gsm

```

5.10 IGameState.hpp

```

00001 #pragma once
00002
00003 #include <memory>
00004 #include "../machine/IGameStateMachine.hpp"
00005
00006 namespace gsm {

```

```

00007
00008 class IGameState {
00009 public:
00010     virtual ~IGameState() = default;
00011
00012     virtual void enter() = 0;
00013     virtual void update(float deltaTime) = 0;
00014     virtual void exit() = 0;
00015 };
00016
00017 } // namespace gsm

```

5.11 IServer.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** IServer
00006 */
00007
00008 #ifndef ISERVER_HPP_
00009 #define ISERVER_HPP_
00010
00011 namespace net {
00012
00013 class IServer {
00014 public:
00015     virtual ~IServer() = default;
00016     virtual void start() = 0;
00017     virtual void stop() = 0;
00018 };
00019
00020 } // namespace net
00021
00022 #endif /* !ISERVER_HPP_ */

```

5.12 Server.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** Server
00006 */
00007
00008 #ifndef SERVER_HPP_
00009 #define SERVER_HPP_
00010
00011 #include "IServer.hpp"
00012
00013 namespace net {
00014
00015 class Server : public IServer {
00016 public:
00017     Server();
00018     ~Server();
00019
00020     void start() override;
00021     void stop() override;
00022
00023 protected:
00024 private:
00025 };
00026
00027 } // namespace net
00028
00029 #endif /* !SERVER_HPP_ */

```

5.13 ServerConfig.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025

```

```
00003 ** ryanR-type
00004 ** File description:
00005 ** ServerConfig
00006 */
00007
00008 #ifndef SERVERCONFIG_HPP_
00009 #define SERVERCONFIG_HPP_
00010
00011 namespace net {
00012
00013 class ServerConfig {
00014     public:
00015         ServerConfig();
00016         ~ServerConfig();
00017
00018     protected:
00019     private:
00020 };
00021
00022 } // namespace net
00023
00024 #endif /* !SERVERCONFIG_HPP_ */
```


Index

/home/albane/epitech/tech3/r-type/ryanR-type/client/graphics/IGameStateMachine, 12
15
/home/albane/epitech/tech3/r-type/ryanR-type/client/graphics/IWindow.hpp,
15
net::Server, 13
/home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/machine/AGameStateMachine.hpp,
16
start, 14
stop, 14
/home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/machine/IGameStateMachine.hpp,
17
net::ServerConfig, 14
/home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/AGameState.hpp,
17
popState, 9
gsm::AGameStateMachine, 9
/home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/IGameState.hpp,
18
pushState, 10
gsm::AGameStateMachine, 10
/home/albane/epitech/tech3/r-type/ryanR-type/server/IServer.hpp,
19
render
/home/albane/epitech/tech3/r-type/ryanR-type/server/Server.hpp,
19
gsm::AGameState, 8
gsm::AGameStateMachine, 10
/home/albane/epitech/tech3/r-type/ryanR-type/server/ServerConfig.hpp,
19
start
/home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/machine/AGameStateMachine.hpp,
16
net::Server, 14
stop
/home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/machine/IGameStateMachine.hpp,
17
net::Server, 14
update
/home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/states/AGameState.hpp,
18
gsm::AGameState, 8
gsm::AGameStateMachine, 10
/home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/states/IGameState.hpp,
18

changeState
gsm::AGameStateMachine, 9

enter
gsm::AGameState, 7

exit
gsm::AGameState, 8

gfx::color_t, 11
gfx::IEvent, 11
gfx::IWindow, 13
gsm::AGameState, 7
enter, 7
exit, 8
render, 8
update, 8
gsm::AGameStateMachine, 9
changeState, 9
popState, 9
pushState, 10
render, 10
update, 10
gsm::IGameState, 11