

RebeliaProgramowalnychZaskroncow

Wygenerowano przez Doxygen 1.8.9

N, 18 sty 2015 12:28:51

Spis treści

1	Indeks hierarchiczny	1
1.1	Hierarchia klas	1
2	Indeks klas	3
2.1	Lista klas	3
3	Dokumentacja klas	5
3.1	Dokumentacja struktury Address	5
3.2	Dokumentacja klasy AddressRegister	5
3.3	Dokumentacja struktury ChatEntryRaw	6
3.4	Dokumentacja struktury ChatHistoryRaw	6
3.5	Dokumentacja klasy ChatRegister	6
3.6	Dokumentacja klasy Client	7
3.7	Dokumentacja struktury client_comparator	8
3.8	Dokumentacja struktury ClientDataRow	8
3.9	Dokumentacja klasy ClientsRegister	8
3.10	Dokumentacja struktury ClientState	9
3.11	Dokumentacja klasy GameLogic	9
3.12	Dokumentacja klasy GameRoom	9
3.13	Dokumentacja struktury GameRoomRaw	10
3.14	Dokumentacja klasy GameRoomsRegister	11
3.15	Dokumentacja klasy GameState	11
3.16	Dokumentacja struktury HandshakeRaw	12
3.17	Dokumentacja klasy Logger	12
3.18	Dokumentacja klasy LuaInterpreter	13
3.19	Dokumentacja klasy Observer	13
3.20	Dokumentacja klasy Packet	13
3.20.1	Dokumentacja składowych wyliczanych	14
3.20.1.1	Tag	14
3.20.2	Dokumentacja funkcji składowych	14
3.20.2.1	serialize	14
3.21	Dokumentacja klasy Packet_handler	15

3.22 Dokumentacja struktury Resource	15
3.23 Dokumentacja struktury ServerResources	16
3.24 Dokumentacja klasy Subject	16
3.25 Dokumentacja klasy TcpConnection	17
3.26 Dokumentacja klasy TcpServer	17
Indeks	19

Rozdział 1

Indeks hierarchiczny

1.1 Hierarchia klas

Ta lista dziedziczenia posortowana jest z grubsza, choć nie całkowicie, alfabetycznie:

Address	5
AddressRegister	5
ChatHistoryRaw	6
client_comparator	8
ClientsRegister	8
ClientState	9
enable_shared_from_this	
TcpConnection	17
GameLogic	9
Logger	12
LuaInterpreter	13
noncopyable	
TcpServer	17
Observer	13
Client	7
GameRoomsRegister	11
Packet	13
Packet_handler	15
Resource	15
ChatEntryRaw	6
ClientDataRow	8
Client	7
GameRoomRaw	10
GameRoom	9
GameState	11
HandshakeRaw	12
ServerResources	16
TcpServer	17
Subject	16
ChatRegister	6
GameRoom	9

Rozdział 2

Indeks klas

2.1 Lista klas

Tutaj znajdują się klasy, struktury, unie i interfejsy wraz z ich krótkimi opisami:

Address	5
AddressRegister	5
ChatEntryRaw	6
ChatHistoryRaw	6
ChatRegister	6
Client	7
client_comparator	8
ClientDataRow	8
ClientsRegister	8
ClientState	9
GameLogic	9
GameRoom	9
GameRoomRaw	10
GameRoomsRegister	11
GameState	11
HandshakeRaw	12
Logger	12
LuaInterpreter	13
Observer	13
Packet	13
Packet_handler	15
Resource	15
ServerResources	16
Subject	16
TcpConnection	17
TcpServer	17

Rozdział 3

Dokumentacja klas

3.1 Dokumentacja struktury Address

Metody publiczne

- **Address** (AddressIP Ip="non-defined", AddressPort Port=~0)
- void **change_owner** (ClientID newOwner) const
- void **update_connection** (const TcpPointer &x) const
- bool **operator<** (const Address &latter) const

Atrybuty publiczne

- AddressIP **ip**
- AddressPort **port**
- ClientID **owner**
- TcpPointer **connection**

Dokumentacja dla tej struktury została wygenerowana z plików:

- src/server/Address.hpp
- src/server/Address.cpp

3.2 Dokumentacja klasy AddressRegister

Metody publiczne

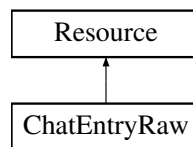
- const Address * **register_address** (const Address &x)
- ClientID **get_address_owner** (const Address &x) const
- const Address * **get_address_pointer** (const Address &x) const

Dokumentacja dla tej klasy została wygenerowana z plików:

- src/server/AddressRegister.hpp
- src/server/AddressRegister.cpp

3.3 Dokumentacja struktury ChatEntryRaw

Diagram dziedziczenia dla ChatEntryRaw



Metody publiczne

- **ChatEntryRaw** (const std::string &nick__, const std::string &message__)
- virtual Resource::Tag **get_tag** ()
- template<class Archive >
void **serialize** (Archive &ar, const unsigned int)
- virtual std::string **show_content** ()

Atrybuty publiczne

- std::string **nick_**
- std::string **message_**

Przyjaciele

- class **boost::serialization::access**

Dodatkowe Dziedziczone Składowe

Dokumentacja dla tej struktury została wygenerowana z pliku:

- src/shared/ChatEntryRaw.hpp

3.4 Dokumentacja struktury ChatHistoryRaw

Przyjaciele

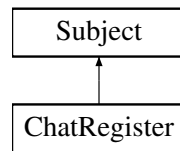
- class **boost::serialization::access**

Dokumentacja dla tej struktury została wygenerowana z pliku:

- src/shared/ChatHistoryRaw.hpp

3.5 Dokumentacja klasy ChatRegister

Diagram dziedziczenia dla ChatRegister



Metody publiczne

- void **register_message** ([ChatEntryRaw](#) &message)

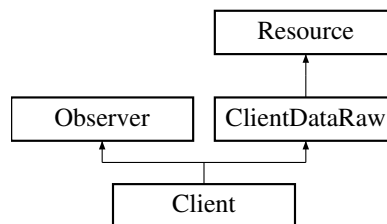
Dodatkowe Dziedziczone Składowe

Dokumentacja dla tej klasy została wygenerowana z plików:

- src/shared/ChatRegister.hpp
- src/shared/ChatRegister.cpp

3.6 Dokumentacja klasy Client

Diagram dziedziczenia dla Client



Metody publiczne

- **Client** (const [Address](#) *address, TcpPointer pointer=nullptr, std::string nick="UNREGISTERED")
- **Client** (const [Client](#) &c)
- ClientID **get_client_id** () const
- [ClientState](#) **get_state** () const
- std::string **get_nickname** () const
- const [Address](#) * **get_address** () const
- void **set_state** ([ClientState](#) s) const
- bool **operator<** (const [Client](#) &) const
- virtual void **update** ([Resource](#) *updateInfo)

Dodatkowe Dziedziczone Składowe

Dokumentacja dla tej klasy została wygenerowana z plików:

- src/server/Client.hpp
- src/server/Client.cpp

3.7 Dokumentacja struktury client_comparator

Metody publiczne

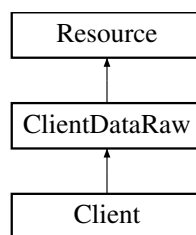
- bool **operator()** (const ClientPtr &a, const ClientPtr &b) const

Dokumentacja dla tej struktury została wygenerowana z pliku:

- src/server/ClientsRegister.hpp

3.8 Dokumentacja struktury ClientDataRaw

Diagram dziedziczenia dla ClientDataRaw



Metody publiczne

- **ClientDataRaw** (ClientID clientID, std::string nickname, [ClientState](#) state)
- virtual Tag **get_tag** ()
- virtual std::string **show_content** ()

Atrybuty publiczne

- [ClientState](#) **state_**
- const ClientID **clientID_**
- std::string **nickname_**

Przyjaciele

- class **boost::serialization::access**

Dodatkowe Dziedziczone Składowe

Dokumentacja dla tej struktury została wygenerowana z pliku:

- src/server/ClientDataRaw.hpp

3.9 Dokumentacja klasy ClientsRegister

Metody publiczne

- ClientIt **register_client** (const [Address](#) *address, TcpPointer pointer)

- ClientIt **look_up_with_id** (ClientID id) const
- void **change_state** (ClientID id, ClientState state)
- void **remove_client** (ClientID id)
- void **remove_client** (ClientIt it)
- ClientState **get_state** (const ClientIt &it) const
- int **get_size** () const

Dokumentacja dla tej klasy została wygenerowana z plików:

- src/server/ClientsRegister.hpp
- src/server/ClientsRegister.cpp

3.10 Dokumentacja struktury ClientState

Typy publiczne

- enum **Location** { **LOBBY**, **GAMEROOM**, **GAME** }

Metody publiczne

- **ClientState** (Location l=LOBBY, int lIdentifier=0)

Atrybuty publiczne

- Location **location**
- int **locationIdentifier**

Przyjaciele

- class **boost::serialization::access**

Dokumentacja dla tej struktury została wygenerowana z pliku:

- src/server/ClientDataRaw.hpp

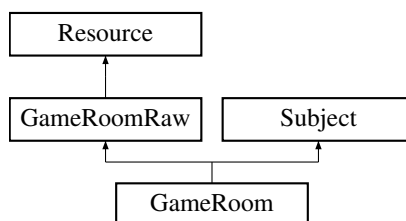
3.11 Dokumentacja klasy GameLogic

Dokumentacja dla tej klasy została wygenerowana z pliku:

- src/shared/GameLogic.hpp

3.12 Dokumentacja klasy GameRoom

Diagram dziedziczenia dla GameRoom



Metody publiczne

- **GameRoom** (ClientID host, std::string gameRoomName)
- virtual void **notify** ()
- void **add_player** (ClientID newPlayer)
- void **remove_player** (ClientID player)
- GameRoomID **get_id** ()
- unsigned int **get_number_of_players** ()

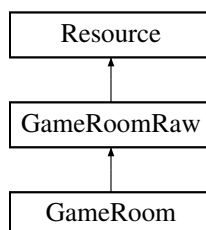
Dodatkowe Dziedziczone Składowe

Dokumentacja dla tej klasy została wygenerowana z plików:

- src/shared/GameRoom.hpp
- src/shared/GameRoom.cpp

3.13 Dokumentacja struktury GameRoomRaw

Diagram dziedziczenia dla GameRoomRaw



Metody publiczne

- **GameRoomRaw** (std::string name, ClientID hostID, GameRoomID GRId)
- virtual Tag **get_tag** ()
- virtual std::string **show_content** ()
- template<class Archive >
void **serialize** (Archive &ar, const unsigned int)

Atrybuty publiczne

- std::string **gameRoomName**
- const ClientID **host**
- std::list< ClientID > **players**
- unsigned int **numOfPlayers**
- unsigned int **maxNumOfPlayers**
- const GameRoomID **id**

Przyjaciele

- class **boost::serialization::access**

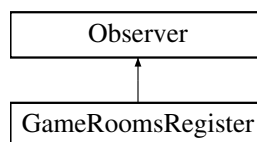
Dodatkowe Dziedziczone Składowe

Dokumentacja dla tej struktury została wygenerowana z plików:

- src/shared/GameRoomRaw.hpp
- src/shared/GameRoomRaw.cpp

3.14 Dokumentacja klasy GameRoomsRegister

Diagram dziedziczenia dla GameRoomsRegister



Metody publiczne

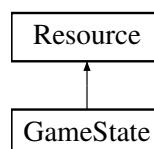
- `GameRoomPtr` **add_game_room** (`ClientID` host, `std::string` name)
- void **remove_game_room** (`GameRoomID` id)
- `GameRoomPtr` **look_up_with_id** (`GameRoomID` id)
- unsigned int **get_size** ()

Dokumentacja dla tej klasy została wygenerowana z plików:

- src/shared/GameRoomsRegister.hpp
- src/shared/GameRoomsRegister.cpp

3.15 Dokumentacja klasy GameState

Diagram dziedziczenia dla GameState



Metody publiczne

- virtual `std::string` **show_content** ()
- virtual `Resource::Tag` **get_tag** ()

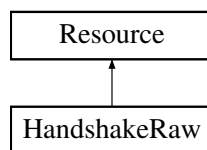
Dodatkowe Dziedziczone Składowe

Dokumentacja dla tej klasy została wygenerowana z plików:

- `src/shared/GameState.hpp`
- `src/shared/GameState.cpp`

3.16 Dokumentacja struktury HandshakeRaw

Diagram dziedziczenia dla HandshakeRaw



Metody publiczne

- **HandshakeRaw** ([Address](#) ad=[Address](#)(0, 0))
- virtual `Resource::Tag` **get_tag** ()
- `template<class Archive >`
void **serialize** (Archive &ar, const unsigned int)
- virtual `std::string` **show_content** ()

Atrybuty publiczne

- [Address](#) **address_**

Przyjaciele

- class **boost::serialization::access**

Dodatkowe Dziedziczone Składowe

Dokumentacja dla tej struktury została wygenerowana z pliku:

- `src/shared/HandshakeRaw.hpp`

3.17 Dokumentacja klasy Logger

Statyczne metody publiczne

- static [Logger](#) & **getInstance** ()
- static `log4cpp::CategoryStream` **log_debug** ()
- static `log4cpp::CategoryStream` **log_main** ()

Atrybuty publiczne

- log4cpp::Category & **main**
- log4cpp::Category & **debug**

Dokumentacja dla tej klasy została wygenerowana z plików:

- src/server/Logger.hpp
- src/server/Logger.cpp

3.18 Dokumentacja klasy LuaInterpreter

Metody publiczne

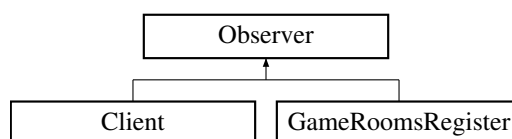
- int **interprete** (QString)
- void **registerFunctions** ()

Dokumentacja dla tej klasy została wygenerowana z plików:

- src/luacore/LuaInterpreter.hpp
- src/luacore/LuaInterpreter.cpp

3.19 Dokumentacja klasy Observer

Diagram dziedziczenia dla Observer



Metody publiczne

- virtual void **update** (Resource *)

Dokumentacja dla tej klasy została wygenerowana z pliku:

- src/shared/Observer.hpp

3.20 Dokumentacja klasy Packet

Typy publiczne

- enum **Tag** {
 UPDATED_RESOURCE, REGISTER_REQUEST, CHAT_ENTRY_MESSAGE_REQUEST, GAMEROOM_CREATE_REQUEST,
 GAMEROOM_JOIN_REQUEST, GAMEROOM_LEAVE_REQUEST, GAMEROOM_UPDATE_REQUEST,
 GAMEROOM_START_REQUEST,
 GAME_START_FAILURE_INFO, SYNCHRONISE_REQUEST, CLOCK_SYNCHRONISE, GAME_STATE,
 GAME_ACTION, KEEP_ALIVE }

Enum będący flagą stanowiącą podstawę komunikacji w naszym systemie.

- typedef std::string **StreamBuffer**

Metody publiczne

- **Packet** ([Tag](#) tag__, const [Address](#) *ad__=nullptr, [Resource](#) *content__=nullptr)
- **Packet** ([Tag](#) tag__, const [Address](#) *ad__, ResourcePtr content__=nullptr)
- template<class Archive >
void [serialize](#) (Archive &ar, const unsigned int)
Serializacja Pakietu dotyczy tagu pakietu i zawartości (wskaźnika na.
- StreamBuffer **get_data_streambuf** ()
- const [Address](#) * **get_address** () const
- [Tag](#) **get_tag** () const
- ResourcePtr **get_content** () const
- std::string **show_resource_content** ()

Przyjaciele

- class **boost::serialization::access**

3.20.1 Dokumentacja składowych wyliczanych

3.20.1.1 enum Packet::Tag

Enum będący flagą stanowiącą podstawę komunikacji w naszym systemie.

Ten enum może urosnąć duży, ponieważ do każdej dodanej funkcjonalności, będzie nowy tag pakietu.

Wartości wyliczeń

UPDATED_RESOURCE dane aktualizacyjne dla klienta, można rozpoznać zawartość po tagu [Resource](#)

REGISTER_REQUEST w środku pakietu Handshake przedstawiający dane o kliencie.

CHAT_ENTRY_MESSAGE_REQUEST prośba o nadanie wiadomości czatu

GAMEROOM_CREATE_REQUEST prośba o stworzenie nowego pokoju

GAMEROOM_JOIN_REQUEST prośba o dołączenie do pokoju

GAMEROOM_LEAVE_REQUEST prośba o opuszczenie pokoju

GAMEROOM_UPDATE_REQUEST prośba o zmianę ustawień pokoju (np. ustawień gry dla hosta, a dla gracza wyrażenie gotowości)

GAMEROOM_START_REQUEST prośba o rozpoczęcie rozgrywki

GAME_START_FAILURE_INFO informacja dla klienta o niespełnionym żądaniu, np. z powodu niegotowości wszystkich graczy

SYNCHRONISE_REQUEST prośba o wysłanie wszystkich zasobów... podejrzewam, że można się bez tego obyć i zamiast tego zrobić timeouty

CLOCK_SYNCHRONISE prośba o określenie czasu względem serwera (służy też do obliczenia round trip time)

GAME_STATE pakiet zawierający stan rozgrywki - do przekierowania dla konkretnej instancji gry

GAME_ACTION być może się przyda?

KEEP_ALIVE ping! do ustalenia czy ktoś stracił połączenie.

3.20.2 Dokumentacja funkcji składowych

3.20.2.1 template<class Archive > void Packet::serialize (Archive & ar, const unsigned int) [inline]

Serializacja Pakietu dotyczy tagu pakietu i zawartości (wskaźnika na.

Zobacz również

[Resource](#))

Dokumentacja dla tej klasy została wygenerowana z plików:

- src/shared/Packet.hpp
- src/shared/Packet.cpp

3.21 Dokumentacja klasy Packet_handler

Metody publiczne

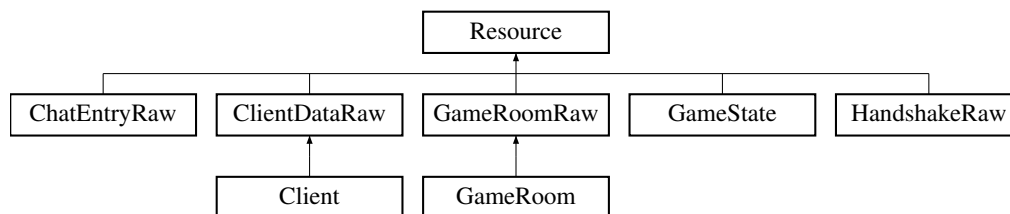
- **Packet_handler** (std::queue< [Packet](#) > *inQueue, bool *running)
- void **operator()** ()

Dokumentacja dla tej klasy została wygenerowana z plików:

- src/server/Packet_handler.hpp
- src/server/Packet_handler.cpp

3.22 Dokumentacja struktury Resource

Diagram dziedziczenia dla Resource



Typy publiczne

- enum **Tag** { CHAT_ENTRY, CLIENT_DATA, GAMEROOM, HANDSHAKE }

Metody publiczne

- virtual Tag **get_tag** ()=0
- template<typename Archive >
void **serialize** (Archive &, const unsigned int)
- virtual std::string **show_content** ()=0

Przyjaciele

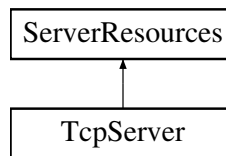
- class **boost::serialization::access**

Dokumentacja dla tej struktury została wygenerowana z pliku:

- src/shared/Resource.hpp

3.23 Dokumentacja struktury ServerResources

Diagram dziedziczenia dla ServerResources



Metody publiczne

- void **init** ()

Atrybuty publiczne

- std::thread * **packetHandler**
- PacketQueue **received**
- [Packet_handler](#) **lobbyManager**
- [ClientsRegister](#) **connectedClients**
- [AddressRegister](#) **registeredAddresses**

Atrybuty chronione

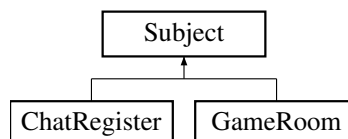
- bool **running_**

Dokumentacja dla tej struktury została wygenerowana z pliku:

- src/server/ServerResources.hpp

3.24 Dokumentacja klasy Subject

Diagram dziedziczenia dla Subject



Metody publiczne

- void **addObserver** ([Observer](#) *o)
- virtual void **notify** ()=0

Atrybuty chronione

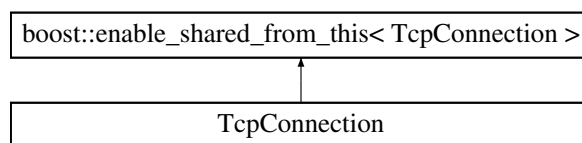
- std::vector< [Observer](#) * > **obs_**

Dokumentacja dla tej klasy została wygenerowana z pliku:

- src/shared/Subject.hpp

3.25 Dokumentacja klasy TcpConnection

Diagram dziedziczenia dla TcpConnection



Metody publiczne

- tcp::socket & **socket** ()
- void **write** (std::string)
- std::string **read** () const
- std::string **ip_address** () const
- void **wait_data** ()
- unsigned short **port** () const

Statyczne metody publiczne

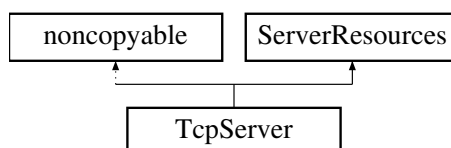
- static TcpPointer **create** (boost::asio::io_service &)

Dokumentacja dla tej klasy została wygenerowana z plików:

- src/shared/TcpConnection.hpp
- src/shared/TcpConnection.cpp

3.26 Dokumentacja klasy TcpServer

Diagram dziedziczenia dla TcpServer



Metody publiczne

- void **start** ()
- void **stop** ()
- unsigned **connections** ()

Statyczne metody publiczne

- static [TcpServer](#) & **getInstance** ()

Dodatkowe Dziedziczone Składowe

Dokumentacja dla tej klasy została wygenerowana z plików:

- `src/server/Server.hpp`
- `src/server/Server.cpp`

Skorowidz

Address, [5](#)
AddressRegister, [5](#)

CHAT_ENTRY_MESSAGE_REQUEST
 Packet, [14](#)
CLOCK_SYNCHRONISE
 Packet, [14](#)
ChatEntryRaw, [6](#)
ChatHistoryRaw, [6](#)
ChatRegister, [6](#)
Client, [7](#)
client_comparator, [8](#)
ClientDataRow, [8](#)
ClientState, [9](#)
ClientsRegister, [8](#)

GAME_ACTION
 Packet, [14](#)
GAME_START_FAILURE_INFO
 Packet, [14](#)
GAME_STATE
 Packet, [14](#)
GAMEROOM_CREATE_REQUEST
 Packet, [14](#)
GAMEROOM_JOIN_REQUEST
 Packet, [14](#)
GAMEROOM_LEAVE_REQUEST
 Packet, [14](#)
GAMEROOM_START_REQUEST
 Packet, [14](#)
GAMEROOM_UPDATE_REQUEST
 Packet, [14](#)
GameLogic, [9](#)
GameRoom, [9](#)
GameRoomRaw, [10](#)
GameRoomsRegister, [11](#)
GameState, [11](#)

HandshakeRaw, [12](#)

KEEP_ALIVE
 Packet, [14](#)

Logger, [12](#)
LuaInterpreter, [13](#)

Observer, [13](#)

Packet, [13](#)
 CHAT_ENTRY_MESSAGE_REQUEST, [14](#)
 CLOCK_SYNCHRONISE, [14](#)
 GAME_ACTION, [14](#)
 GAME_START_FAILURE_INFO, [14](#)
 GAME_STATE, [14](#)
 GAMEROOM_CREATE_REQUEST, [14](#)
 GAMEROOM_JOIN_REQUEST, [14](#)
 GAMEROOM_LEAVE_REQUEST, [14](#)
 GAMEROOM_START_REQUEST, [14](#)
 GAMEROOM_UPDATE_REQUEST, [14](#)
 KEEP_ALIVE, [14](#)
 REGISTER_REQUEST, [14](#)
 SYNCHRONISE_REQUEST, [14](#)
 serialize, [14](#)
 Tag, [14](#)
 UPDATED_RESOURCE, [14](#)
Packet_handler, [15](#)

REGISTER_REQUEST
 Packet, [14](#)
Resource, [15](#)

SYNCHRONISE_REQUEST
 Packet, [14](#)
serialize
 Packet, [14](#)
ServerResources, [16](#)
Subject, [16](#)

Tag
 Packet, [14](#)
TcpConnection, [17](#)
TcpServer, [17](#)

UPDATED_RESOURCE
 Packet, [14](#)