My Project

Generated by Doxygen 1.8.3.1

Tue Dec 3 2013 02:22:03

Contents

1	Hier	archical Index	1
	1.1	Class Hierarchy	1
2	Clas	s Index	3
	2.1	Class List	3
3	Clas	s Documentation	5
	3.1	Bar Class Reference	5
	3.2	Button Class Reference	5
	3.3	CandyFactory Class Reference	6
	3.4	Caramelo Class Reference	6
		3.4.1 Member Function Documentation	7
		3.4.1.1 mover	7
	3.5	Cliente Class Reference	7
		3.5.1 Detailed Description	7
		3.5.2 Member Function Documentation	7
		3.5.2.1 mostrarVentanalP	7
	3.6	ClienteInterface Class Reference	8
		3.6.1 Detailed Description	8
		3.6.2 Member Function Documentation	8
		3.6.2.1 nuevoMensaje	8
	3.7	GameWindow Class Reference	8
		3.7.1 Detailed Description	9
		3.7.2 Member Function Documentation	9
		3.7.2.1 mensaje	9
		3.7.2.2 on_mensaje	9
	3.8	Ipwindow Class Reference	10
		3.8.1 Detailed Description	10
		3.8.2 Member Typedef Documentation	11
		3.8.2.1 type_signal_conectar	11
		3.8.3 Member Function Documentation	11
		3.8.3.1 mensaje	11

ii CONTENTS

3.9	Listado	r Class Reference	11
	3.9.1	Detailed Description	11
	3.9.2	Member Function Documentation	11
		3.9.2.1 listar	11
3.10	Logger	Class Reference	11
	3.10.1	Detailed Description	12
	3.10.2	Member Function Documentation	12
		3.10.2.1 log	12
3.11	MainW	indow Class Reference	12
	3.11.1	Detailed Description	13
	3.11.2	Member Function Documentation	13
		3.11.2.1 mensaje	13
3.12	Mutex	Class Reference	13
	3.12.1	Detailed Description	14
	3.12.2	Member Function Documentation	14
		3.12.2.1 lock	14
		3.12.2.2 unlock	14
3.13	Partida	Class Reference	14
	3.13.1	Detailed Description	15
	3.13.2	Constructor & Destructor Documentation	15
		3.13.2.1 Partida	15
	3.13.3	Member Function Documentation	15
		3.13.3.1 addUsuario	15
		3.13.3.2 broadcastMsj	15
		3.13.3.3 getNivel	15
		3.13.3.4 mensaje	16
		3.13.3.5 msjError	16
		3.13.3.6 rmUsuario	16
3.14	Partida	Interface Class Reference	16
	3.14.1	Detailed Description	17
	3.14.2	Member Function Documentation	17
		3.14.2.1 mensaje	17
3.15	Server	Class Reference	17
	3.15.1	Detailed Description	18
	3.15.2	Member Function Documentation	18
		3.15.2.1 listPartidas	18
		3.15.2.2 removePartida	18
	3.15.3	Member Data Documentation	18
			18
3.16	Serverl	Interface Class Reference	18

CONTENTS

3.17	Socket	Class Reference	19
3.18	Socket	O Class Reference	19
	3.18.1	Detailed Description	20
	3.18.2	Member Function Documentation	20
		3.18.2.1 read	20
		3.18.2.2 write	20
3.19	Sound	Player Class Reference	20
	3.19.1	Detailed Description	21
	3.19.2	Member Function Documentation	21
		3.19.2.1 play	21
	3.19.3	Member Data Documentation	21
		3.19.3.1 archs	21
3.20	Star Cl	ass Reference	21
3.21	Tablero	Class Reference	22
	3.21.1	Member Function Documentation	23
		3.21.1.1 activarCombinacionColumna	23
		3.21.1.2 activarCombinacionColumna	23
		3.21.1.3 activarCombinacionFila	23
		3.21.1.4 activarCombinacionFila	24
		3.21.1.5 calcularCoordenadas	24
		3.21.1.6 dispararColumna	24
		3.21.1.7 dispararFila	24
		3.21.1.8 doCombinacion	25
		3.21.1.9 doStar	25
		3.21.1.10 esButton	25
		3.21.1.11 esMismoColor	25
		3.21.1.12 hayMovimiento	25
		3.21.1.13 hayMovimientos	25
		3.21.1.14 hayPatrones	26
		3.21.1.15 horBarColor	26
		3.21.1.16 movimiento	26
		3.21.1.17 movimientoValido	26
		3.21.1.18 movInverso	26
		3.21.1.19 newMov	27
		3.21.1.20 newMov	27
		3.21.1.21 num2str	27
		3.21.1.22 rellenarTablero	27
		3.21.1.23 verBarColor	28
3.22	Tablero	Juego Class Reference	28
	3.22.1		28

iv CONTENTS

	3.22.1.1 mensaje	28
3.23	TCPSocket Class Reference	28
3.24	TCPSocketConnect Class Reference	29
	3.24.1 Detailed Description	29
3.25	TCPSocketListener Class Reference	29
	3.25.1 Member Function Documentation	30
	3.25.1.1 accept	30
	3.25.1.2 listen	30
3.26	Thread Class Reference	30
3.27	ThreadListener Class Reference	31
	3.27.1 Detailed Description	31
	3.27.2 Member Function Documentation	32
	3.27.2.1 setKey	32
3.28	ThreadSocket Class Reference	32
	3.28.1 Detailed Description	32
3.29	ThreadUsuario Class Reference	33
	3.29.1 Detailed Description	33
3.30	TWavArgs Struct Reference	34
3.31	TWavHeader Struct Reference	34
3.32	UserManager Class Reference	34
	3.32.1 Detailed Description	35
	3.32.2 Member Function Documentation	35
	3.32.2.1 get	35
	3.32.2.2 set	35
3.33	Window Class Reference	35
	3.33.1 Detailed Description	36
	3.33.2 Member Typedef Documentation	36
	3.33.2.1 type_signal_mensaje	36
	3.33.3 Member Function Documentation	36
	3.33.3.1 mensaje	36

Index

36

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Button	
Caramelo	6
Bar	5
Button	5
Star	21
CandyFactory	6
ClienteInterface	8
Cliente	7
	11
	11
	13
	16
	14
	18
	17
	19
	19
	29
TCPSocket	
TCPSocketConnect	_
TCPSocketListener	
	20
Tablero	22
Thread	30
Server	17
ThreadSocket	32
ThreadListener	31
ThreadUsuario	33
TWavArgs	34
TWavHeader	34
UserManager	34
Window	
Window	35
GameWindow	8
Ipwindow	10
MainWindow	12
TableroJuego	28

2 **Hierarchical Index**

Chapter 2

Class Index

2.1 Class List

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

Bar	5
Button	5
CandyFactory	6
Caramelo	6
Cliente	7
ClienteInterface	8
GameWindow	8
	10
Listador	11
	11
MainWindow	12
Mutex	13
Partida 1	14
PartidaInterface	16
Server	17
ServerInterface	18
Socket	19
SocketIO 1	19
SoundPlayer	20
Star	21
Tablero	22
TableroJuego	28
TCPSocket	28
TCPSocketConnect	29
TCPSocketListener	29
Thread	30
ThreadListener	31
ThreadSocket	32
ThreadUsuario	33
TWavArgs	34
TWavHeader	34
UserManager	34
	2=

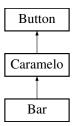
Class Index

Chapter 3

Class Documentation

3.1 Bar Class Reference

Inheritance diagram for Bar:



Public Member Functions

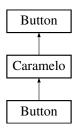
• Bar (int idCaramelo, bool acostado, const std::string &imageName, int i, int j)

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

· cliente.bar.h

3.2 Button Class Reference

Inheritance diagram for Button:



Public Member Functions

• Button (int idCaramelo, const std::string &imageName, int i, int j)

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

· cliente.button.h

3.3 CandyFactory Class Reference

Static Public Member Functions

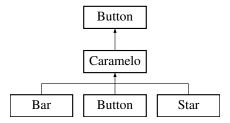
• static Caramelo * crearCaramelo (int id, int i, int j)

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

· cliente.factory_caramelos.h

3.4 Caramelo Class Reference

Inheritance diagram for Caramelo:



Public Member Functions

- · Caramelo (int idCaramelo, const std::string &imageName, int i, int j)
- bool mover (int x, int y)
- bool mover ()
- bool **setMoviendo** (bool moviendo, int x, int y)
- bool setMoviendo (bool moviendo)
- void hablar ()
- int getId ()
- void opacar ()
- · bool visible ()
- bool fullyVisible ()
- void hacerAparecer ()
- int getX ()
- int getXPos ()
- int getY ()
- int getYPos ()
- void setX (int x)
- void setXPos (int x)
- void setY (int y)
- void setYPos (int y)

3.5 Cliente Class Reference 7

3.4.1 Member Function Documentation

3.4.1.1 bool Caramelo::mover (int x, int y)

Chequea si el caramelo esta en movimiento. Si esta en movimiento, pero se pasan como parametros las coordenadas finales, devolvera false.

Parameters

X,:	coordenada final x
у,:	coordeanda final y

Returns

verdadero si esta en movmiento, falso si no

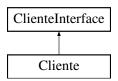
The documentation for this class was generated from the following files:

- · cliente.caramelo.h
- · cliente.caramelo.cpp

3.5 Cliente Class Reference

#include <cliente.cliente.h>

Inheritance diagram for Cliente:



Public Member Functions

void mostrarVentanaIP ()

3.5.1 Detailed Description

Clase cliente. Viene a ser la entidad que representa al programa.

3.5.2 Member Function Documentation

3.5.2.1 void Cliente::mostrarVentanalP()

Muestra la ventana de lp y login.

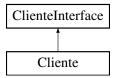
The documentation for this class was generated from the following files:

- · cliente.cliente.h
- · cliente.cliente.cpp

3.6 ClienteInterface Class Reference

#include <cliente.cliente_interface.h>

Inheritance diagram for ClienteInterface:



Public Member Functions

• virtual void nuevoMensaje (Json::Value &msj)=0

3.6.1 Detailed Description

Interfaz de cliente. Es usada por ThreadListener para evitar la referencia circular con cliente.

3.6.2 Member Function Documentation

3.6.2.1 virtual void ClienteInterface::nuevoMensaje (Json::Value & *msj*) [pure virtual]

Agrega un nuevo mensaje a la cola de mensajes (bloque el mutex de la cola)

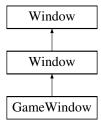
The documentation for this class was generated from the following file:

· cliente.cliente_interface.h

3.7 GameWindow Class Reference

#include <cliente.game.h>

Inheritance diagram for GameWindow:



Public Member Functions

virtual void mensaje (Json::Value &data)

Protected Member Functions

• void on_mensaje ()

- void on_start_game ()
- void on_salir_game ()
- void on_tablero_mensaje (Json::Value data)
- void on_salir ()
- void on_desconectar ()
- virtual bool onClose ()

Protected Attributes

- MenuBarDisconnect menubar
- Gtk::VBox m_VBox
- Gtk::HBox padBox
- Gtk::VBox mainV
- Gtk::ScrolledWindow m ScrolledWindow1
- Gtk::TextView m_TextView1
- Gtk::HBox m_HBox
- Gtk::Entry text_input
- Glib::RefPtr< Gtk::TextBuffer > m_refTextBuffer1
- Gtk::Button m_button_send
- ListaUsuarios user_list
- Gtk::HBox but_hbox
- Gtk::Button button_start
- Gtk::Button button_salir
- TableroJuego * tableroJuego

Additional Inherited Members

3.7.1 Detailed Description

Ventana de espera antes de que se mande la partida a jugar. Basicamente es un chat

3.7.2 Member Function Documentation

```
3.7.2.1 void GameWindow::mensaje ( Json::Value & data ) [virtual]
```

Metodo que reciben los mensajes del servidor, que deben mostrar.

Implements Window.

```
3.7.2.2 void GameWindow::on_mensaje( ) [protected]
```

Signal handler del click en el boton de send

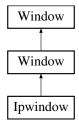
The documentation for this class was generated from the following files:

- · cliente.game.h
- · cliente.game.cpp

3.8 Ipwindow Class Reference

#include <cliente.ipwindow.h>

Inheritance diagram for Ipwindow:



Public Types

 typedef sigc::signal < void, std::string, std::string, std::string, bool > type_signal_conectar

Public Member Functions

- type_signal_conectar signal_conectar ()
- void set_editable (bool is_editable)
- void **set_text** (std::string &str)
- virtual void mensaje (Json::Value &data)

Protected Member Functions

- void on_button_conectar ()
- virtual bool onClose ()

Protected Attributes

- Gtk::Box m HBox
- Gtk::Box m_VBox
- · MenuBar menubar
- Gtk::Image img
- LabelEntry m_host
- LabelEntry m_user
- LabelEntry m_pass
- Gtk::Label m_text
- Gtk::CheckButton m check
- Gtk::HBox m_Button_box
- Gtk::Button m_Button_conectar
- type_signal_conectar m_signal_conectar

3.8.1 Detailed Description

Ventana de login y seleccionar servidor.

3.8.2 Member Typedef Documentation

3.8.2.1 typedef sigc::signal < void, std::string, std::string, bool > lpwindow::type_signal_conectar

Signal para que el Cliente realice la coneccion con el servidor

3.8.3 Member Function Documentation

```
3.8.3.1 void lpwindow::mensaje ( Json::Value & data ) [virtual]
```

Metodo que reciben los mensajes del servidor, que deben mostrar.

Implements Window.

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

- · cliente.ipwindow.h
- · cliente.ipwindow.cpp

3.9 Listador Class Reference

```
#include <server.listador.h>
```

Static Public Member Functions

- static Json::Value listar ()
- static int **getNivel** (std::string &fileName)
- static int getMapa (std::string &fileName, Json::Value &mapa)

3.9.1 Detailed Description

Clase que se encarga de proveer metodos para traer informacion acerca de los mapas.

3.9.2 Member Function Documentation

```
3.9.2.1 static Json::Value Listador::listar() [inline], [static]
```

Devuelve un objeto (hash) con: { nombre de archivo : nivel del mapa, }

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

· server.listador.h

3.10 Logger Class Reference

```
#include <common.logger.h>
```

Static Public Member Functions

- static void init ()
- static void destroy ()

- static void log (const std::string &str)
- static void log (const char *str)

Protected Member Functions

· void print (const std::string &str)

Protected Attributes

Mutex mut

Static Protected Attributes

• static Logger * me = NULL

3.10.1 Detailed Description

Clase singleton usada para logger. Bloque mutex antes de escribir al stdout, se la debe iniciar con init y destruir con destroy.

3.10.2 Member Function Documentation

3.10.2.1 static void Logger::log (const std::string & str) [static]

Funciones usadas para logear.

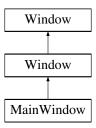
The documentation for this class was generated from the following files:

- · common.logger.h
- · common.logger.cpp

3.11 MainWindow Class Reference

#include <cliente.main_window.h>

Inheritance diagram for MainWindow:



Public Member Functions

- void **setText** (std::string &str)
- virtual void mensaje (Json::Value &data)

3.12 Mutex Class Reference 13

Protected Member Functions

- void onListGames (int code, Json::Value &data)
- void onListMaps (int code, Json::Value &data)
- void on_partidas ()
- void join_partidas ()
- void on_mapas ()
- void on crear partida ()
- virtual bool onClose ()

Protected Attributes

- MenuBarDisconnect menubar
- Gtk::Notebook tabs
- Gtk::Label labelPartidas
- Gtk::VBox m_VBox_partidas
- Gtk::HBox m HBox partidas buttons
- Gtk::Button button_partidas_act
- Gtk::Button button_partidas_con
- Gtk::ScrolledWindow m_ScrolledPartidas
- ListaPartidas m_TreeView
- Gtk::Label labelMapas
- Gtk::VBox m_VBox_mapas
- Gtk::HBox m_HBox_mapas_buttons
- Gtk::Button button_mapas_act
- · Gtk::Button button_mapas_cre
- Gtk::ScrolledWindow m_ScrolledMapas
- ListaMapas m_TreeViewMapas
- Gtk::VBox mainV
- Gtk::HBox tabBox
- · Gtk::Label statusLabel

Additional Inherited Members

3.11.1 Detailed Description

Ventana de Seleccionar y/o crear partida

3.11.2 Member Function Documentation

3.11.2.1 void MainWindow::mensaje (Json::Value & data) [virtual]

Metodo que reciben los mensajes del servidor, que deben mostrar.

Implements Window.

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

- · cliente.main window.h
- · cliente.main_window.cpp

3.12 Mutex Class Reference

#include <common.mutex.h>

Public Member Functions

- int lock ()
- int unlock ()

3.12.1 Detailed Description

Encapsulamiento del mutex.

3.12.2 Member Function Documentation

```
3.12.2.1 int Mutex::lock ( )
```

Lockea al mutex

```
3.12.2.2 int Mutex::unlock ( )
```

Desloquea al mutex

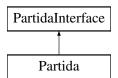
The documentation for this class was generated from the following files:

- · common.mutex.h
- · common.mutex.cpp

3.13 Partida Class Reference

```
#include <server.partida.h>
```

Inheritance diagram for Partida:



Public Member Functions

- Partida (ServerInterface *server, int nivel, std::string &nombre)
- void addUsuario (ThreadSocket *u, Json::Value &user)
- void rmUsuario (ThreadSocket *u)
- int getNivel ()
- int getUsuarios ()
- int getMaxUsuarios ()
- PartidaEstado getEstado ()
- std::string getNombre ()
- virtual int mensaje (Json::Value &m, ThreadSocket *u)

Protected Member Functions

- void broadcastMsj (Json::Value &msj)
- void msjError (ThreadSocket *u, const char *msj)
- void msjError (ThreadSocket *u, const std::string &msj)

Protected Attributes

- ServerInterface * server
- std::vector< ThreadSocket *> usuarios
- std::vector< Json::Value > usuarios_data
- int nivel
- std::string nombre
- Mutex usuariosLock
- PartidaEstado estado
- · Json::Value mapa
- int maxUsuarios
- Mutex tableroLock
- Tablero * tablero
- int puntosMax

3.13.1 Detailed Description

Entidad que representa una partida.

3.13.2 Constructor & Destructor Documentation

3.13.2.1 Partida::Partida (ServerInterface * server, int nivel, std::string & nombre)

Creador de la partida.

Parameters

server[in],:	instancia de servidor	
nivel[in],:	nivel de la partida	
nombre[in],:	nombre del mapa	

3.13.3 Member Function Documentation

3.13.3.1 void Partida::addUsuario (ThreadSocket * u, Json::Value & user) [virtual]

Agrega un usuario a la partida.

Parameters

u[in],:	threadsocket del usuario
user[in],: nombre de usuario	

Implements PartidaInterface.

3.13.3.2 void Partida::broadcastMsj (Json::Value & msj) [protected]

Envia msj a todos los usuarios de la partida

3.13.3.3 int Partida::getNivel() [virtual]

Metodos que devuelven informacion acerca de la partida

Implements PartidaInterface.

3.13.3.4 int Partida::mensaje (Json::Value & m, ThreadSocket * u) [virtual]

Metodo usado para mandarle mensajes a la partida.

Parameters

m[in],:	mensaje encondeado en JSON que recibe la partida	
u[in],: ThreadSocket que le envio el mensaje		

Implements PartidaInterface.

```
3.13.3.5 void Partida::msjError ( ThreadSocket * u, const char * msj ) [protected]
```

Envia un mensaje a u, (EVENT_GAME_MSG).

```
3.13.3.6 void Partida::rmUsuario ( ThreadSocket * u ) [virtual]
```

Borra un usuario de la partida

Parameters

```
u[in],: threasocket del usuario
```

Implements PartidaInterface.

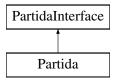
The documentation for this class was generated from the following files:

- · server.partida.h
- · server.partida.cpp

3.14 PartidaInterface Class Reference

```
#include <server.partida_interface.h>
```

Inheritance diagram for PartidaInterface:



Public Member Functions

- virtual void addUsuario (ThreadSocket *u, Json::Value &user)=0
- virtual void **rmUsuario** (ThreadSocket *u)=0
- virtual int mensaje (Json::Value &m, ThreadSocket *u)=0
- virtual int **getNivel** ()=0
- virtual int getUsuarios ()=0
- virtual int getMaxUsuarios ()=0
- virtual PartidaEstado getEstado ()=0
- virtual std::string getNombre ()=0

3.14.1 Detailed Description

Interface de partida, para evitar referencia circular.

3.14.2 Member Function Documentation

3.14.2.1 virtual int Partidalnterface::mensaje (Json::Value & m, ThreadSocket * u) [pure virtual]

Se le pasan todos los EVENT_GAME_MISC. Devuelve 0 si todo bien, o 1, si el usuario se desconecto (saca usuario->partida). La parida se debe encargar de toda la borrada de datos de si misma.

Implemented in Partida.

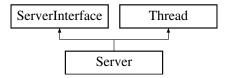
The documentation for this class was generated from the following file:

· server.partida_interface.h

3.15 Server Class Reference

#include <server.server.h>

Inheritance diagram for Server:



Public Member Functions

- Server (int port)
- void removeClient (ThreadSocket *cli)
- PartidaInterface * newPartida (int nivel, std::string &nombre)
- void removePartida (PartidaInterface *p)
- virtual void listPartidas (int nivel, Json::Value &parts)
- virtual PartidaInterface * connectPartidas (long id)
- void **end** ()

Protected Member Functions

- void addClient (ThreadUsuario *cli)
- int main ()
- virtual void * run ()

Protected Attributes

- int port
- std::vector< ThreadUsuario * > clientes
- std::vector< Partida * > partidas
- TCPSocketListener sock
- Mutex clientesLock
- Mutex partidasLock

3.15.1 Detailed Description

Clase de servidor. Esta bloqueada escuchando, cada vez que se conecta un cliente se lanza un nuevo thread con el socket creado (se lanza un ThreadUsuario)

3.15.2 Member Function Documentation

```
3.15.2.1 void Server::listPartidas (int nivel, Json::Value & parts ) [virtual]
```

Devuelve un JSON con las partidas de nivel menor o igual al especificado.

Implements ServerInterface.

```
3.15.2.2 void Server::removePartida ( PartidaInterface * p ) [virtual]
```

Remueve partida de la lista

Implements ServerInterface.

3.15.3 Member Data Documentation

```
3.15.3.1 std::vector<ThreadUsuario*> Server::clientes [protected]
```

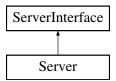
Guarda todos los threads de los usuarios corriendo

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

- · server.server.h
- · server.server.cpp

3.16 ServerInterface Class Reference

Inheritance diagram for ServerInterface:



Public Member Functions

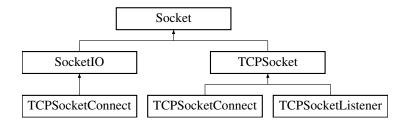
- virtual void removeClient (ThreadSocket *cli)=0
- virtual PartidaInterface * newPartida (int nivel, std::string &nombre)=0
- virtual void removePartida (PartidaInterface *p)=0
- virtual void listPartidas (int nivel, Json::Value &parts)=0
- virtual PartidaInterface * connectPartidas (long id)=0

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

· server.server_interface.h

3.17 Socket Class Reference

Inheritance diagram for Socket:



Public Member Functions

- int shutdown ()
- int shutdown (int how)

Protected Member Functions

- struct sockaddr_in * ip2struct (const int port, const std::string &ip)
- struct sockaddr_in * ip2struct (const std::string &port, const std::string &ip)
- struct sockaddr_in * ip2struct (const std::string &ipport)

Protected Attributes

· unsigned int fd

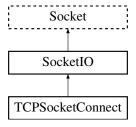
The documentation for this class was generated from the following files:

- · common.socket.h
- · common.socket.cpp

3.18 SocketIO Class Reference

#include <common.socket_io.h>

Inheritance diagram for SocketIO:



Public Member Functions

- · SocketIO (unsigned int fd)
- int read (Json::Value &data, const std::string &key, bool check=true)
- int write (const Json::Value &data, const std::string &key)

Additional Inherited Members

3.18.1 Detailed Description

Socket de entrada y salida. Implementa metodos de escribir y leer con firma.

3.18.2 Member Function Documentation

3.18.2.1 int SocketlO::read (Json::Value & data, const std::string & key, bool check = true)

read: lee mensaje. Lee mensaje de socket. Primero lee prefijo de longitud, dependiendo de eso, lee el mensaje, y despues lee la firma, la validacion de la misma es opcional, pero siempre se la leera. El metodo se encarga de la transformacion de la data json entrante en objeto. En el caso de haber una firma invalida, se saldra con error, pero se devolvera por data el json recibido.

Parameters

out	data	informacion proveniente del socket.
in	key	llave usada para la comprobacion de la firma.
in	check	opcional, (true por defecto) si es verdadero se chequeara la firma.

Returns

0 ok, -1 para cualqier error de lectura de socket, 1 para error de validacion de firma, 2 para problemas de parseo de json.

3.18.2.2 int SocketIO::write (const Json::Value & data, const std::string & key)

write: envia el mensaje. Agrega prefijo de longitud (en bytes) y firma del mensaje. El prefijo de longitud es un uint32_t que esta puesto con hton, despues viene el mensaje (en json) y por ultimo la firma del mensaje

See Also

hmac_msje(). El largo se refiere solo al largo del mensaje, no incluye el largo de la firma que siempre sera el mismo. El orden es cprefijo_longitud><mensaje>cfirma>. El metodo se encarga de serializar.

Parameters

in	data	data a enviar
in	key	llave para la firma

Returns

codigo de error, 0 ok, resto error.

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

- · common.socket io.h
- · common.socket io.cpp

3.19 SoundPlayer Class Reference

#include <cliente.sound_player.h>

3.20 Star Class Reference 21

Static Public Member Functions

- static bool play (const std::string &str)
- static bool play (const char *str)

Protected Member Functions

• void thread_play (int fd, int sample_rate, int seconds)

Static Protected Member Functions

- static bool play_wave (const std::string &str)
- static void * wav_runner (void *arg)

Static Protected Attributes

 static std::map< std::string, std::time_t > archs

3.19.1 Detailed Description

Clase singleton usada para reproducir sonidos. No es threadsafe, solo se puede ejecutar desde un thread. Si se mandan a reproducir dos sonidos iguales en el mismo segundo, solo reproduce el primero.

3.19.2 Member Function Documentation

3.19.2.1 bool SoundPlayer::play (const std::string & str) [static]

Funciones usadas para reproducir.

3.19.3 Member Data Documentation

```
3.19.3.1 std::map < std::string, std::time_t > SoundPlayer::archs [static], [protected]
```

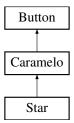
Guarda usando de clave el path, la hora de cunado fue reproducido. Se usa para ignorar archivos iguales si se reproducen en el mismo momento

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

- · cliente.sound_player.h
- cliente.sound_player.cpp

3.20 Star Class Reference

Inheritance diagram for Star:



Public Member Functions

- Star (int idCaramelo, const std::string &imgDir, int i, int j)
- bool mover ()

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

· cliente.star.h

3.21 Tablero Class Reference

Public Member Functions

- Tablero (std::string path)
- int movimiento (int x, int y, CaramelosMovimientos mov, Json::Value &movimientos, int maxPuntos)
- bool hayMovimientos ()
- Json::Value getTablero ()

Protected Member Functions

- void generar ()
- void efectivizarCelda (Json::Value &celda)
- void getMax (double *arr, double &max, int &pos)
- bool movimiento Valido (int x, int y, Caramelos Movimientos mov)
- bool hayMovimiento (int x, int y, Caramelos car)
- bool hayCombinacion (bool todosButtons, int contador)
- void calcularCoordenadas (int x, int y, CaramelosMovimientos mov, int &xf, int &yf)
- int doStar (Caramelos carameloMovido, int x, int y, Caramelos carameloMovido2, int xf, int yf, Json::Value &movimientos)
- bool esMismoColor (Caramelos car, Caramelos car2)
- std::string num2str (int n)
- Json::Value newMov (int x, int y, CaramelosMovimientos mov)
- Json::Value newMov (int x, int y, CaramelosMovimientos mov, Caramelos car)
- CaramelosMovimientos movInverso (CaramelosMovimientos mov)
- void rellenarTablero (Json::Value &movimientos)
- bool hayPatrones (Json::Value &movimientos, int &puntos)
- bool esButton (Caramelos car)
- bool activarCombinacionColumna (int x, int y, bool todosButtons, int contador, Json::Value &movimientos, int &puntos, int x mov, int y mov)
- bool activarCombinacionColumna (int x, int y, bool todosButtons, int contador, Json::Value &movimientos, int &puntos)
- bool activarCombinacionFila (int x, int y, bool todosButtons, int contador, Json::Value &movimientos, int &puntos, int x_mov, int y_mov)

- bool activarCombinacionFila (int x, int y, bool todosButtons, int contador, Json::Value &movimientos, int &puntos)
- void dispararColumna (int x, Json::Value &movimientos, int puntosx, int &puntos)
- void dispararFila (int y, Json::Value &movimientos, int puntosx, int &puntos)
- int doCombinacion (Caramelos car, int x, int y, Json::Value &movimientos)
- Caramelos verBarColor (Caramelos car)
- · Caramelos horBarColor (Caramelos car)

Protected Attributes

- · Json::Value mapa
- Json::Value imagenFondo
- · Json::Value tablero
- · Json::Value probabilidades
- · std::string nMapa
- std::string pathFondo
- int dim_x
- · int dim y
- · int alto
- · int ancho

3.21.1 Member Function Documentation

3.21.1.1 bool Tablero::activarCombinacionColumna (int x, int y, bool todosButtons, int contador, Json::Value & movimientos, int & puntos, int x_mov, int y_mov) [protected]

Activa una combinacion de columna encontrada.

Parameters

x[in],:	
у,:	
todosButtons,:	verdadero si la combinacion es de todos buttons
	array que contiene los movimientos que se realizaron
movimientos[out],-	
:	
puntos[out],:	se le suman los puntos que se ganaron por activar
x_mov,:	posicion x donde se colocara el caramelo especial de tenerse qe generar
y_mov,:	posicion y donde se colocara el caramelo especial de tenerse que generar

Returns

verdadero si se realizo una combinacion

3.21.1.2 bool Tablero::activarCombinacionColumna (int x, int y, bool todosButtons, int contador, Json::Value & movimientos, int & puntos) [protected]

Ver la otra declaracion. Aca se pasan como $x \mod x$ $\mod x$; $y \mod y$ - contador/2;

3.21.1.3 bool Tablero::activarCombinacionFila (int x, int y, bool todosButtons, int contador, Json::Value & movimientos, int & puntos, int x_mov, int y_mov) [protected]

Activa una combinacion de fila encontrada.

Parameters

x[in],:	
у,:	
todosButtons,:	verdadero si la combinacion es de todos buttons
	array que contiene los movimientos que se realizaron
movimientos[out],-	
:	
puntos[out],:	se le suman los puntos que se ganaron por activar
x_mov,:	posicion x donde se colocara el caramelo especial de tenerse qe generar
y_mov,:	posicion y donde se colocara el caramelo especial de tenerse que generar

Returns

verdadero si se realizo una combinacion

3.21.1.4 bool Tablero::activarCombinacionFila (int x, int y, bool todosButtons, int contador, Json::Value & movimientos, int & puntos) [protected]

Ver la otra declaracion. Aca se pasan como x_mov = x - contador/2; y_mov = y;

3.21.1.5 void Tablero::calcularCoordenadas (int x, int y, CaramelosMovimientos mov, int & xf, int & yf) [protected]

Calcula las nuevas coordenadas dsp del movimiento

3.21.1.6 void Tablero::dispararColumna (int x, Json::Value & movimientos, int puntosx, int & puntos) [protected]

Metodo usado para cuando se dispara un bar (para efectuar sobre toda la columna)

Parameters

X,:	posicion x de la columna
	lista de movimientos realizados
movimientos[out],-	
:	
puntosx,:	puntos por cada caramelo sacado
puntos[out],:	se le suman todos los puntos ganados

3.21.1.7 void Tablero::dispararFila (int y, Json::Value & movimientos, int puntosx, int & puntos) [protected]

Metodo usado para cuando se dispara un bar (para efectuar sobre toda la fila)

Parameters

у,:	posicion y de la fila
	lista de movimientos realizados
movimientos[out],-	
:	
puntosx,:	puntos por cada caramelo sacado
puntos[out],:	se le suman todos los puntos ganados

3.21.1.8 int Tablero::doCombinacion (Caramelos car, int x, int y, Json::Value & movimientos) [protected]

Comprueba si hay y efectua la combinacion sobre una ficha.

Parameters

car,:	Caramelo de la ficha
<i>x,:</i>	posicion x del caramelo
у,:	posicion y del caramelo
	lista de movimientos realizados
movimientos[out],-	
:	

Returns

cantidad de puntos sumados

3.21.1.9 int Tablero::doStar (Caramelos *carameloMovido*, int *x*, int *y*, Caramelos *carameloMovido*2, int *xf*, int *yf*, Json::Value & movimientos) [protected]

Efectua accion de intercambio con star.

3.21.1.10 bool Tablero::esButton (Caramelos car) [protected]

Devuelve true si es button

3.21.1.11 bool Tablero::esMismoColor (Caramelos car, Caramelos car2) [protected]

Calcula si las dos fichas son del mismo color.

Parameters

car,:	caramelo
car,:	otro caramelo

Returns

verdadero si son del mismo color, falso si no lo son

3.21.1.12 bool Tablero::hayMovimiento (int x, int y, Caramelos car) [protected]

Chequea qe en la posicion x, y tenga una combinacion a su alrededor

Returns

true si hay, false si no

3.21.1.13 bool Tablero::hayMovimientos ()

Devuelve verdadero si hay movimientos para realizar

3.21.1.14 bool Tablero::hayPatrones (Json::Value & movimientos, int & puntos) [protected]

Busca si hay patrones y los reemplaza por RELLENAR, sumandole a puntos los puntos ganados. Puntos debe estar inicializado previamente en algun valor.

Parameters

	movimientos que se realizan
movimientos[out],-	
:	
puntos[out],:	puntos ganados

Returns

true si hubo patrones, false si no.

3.21.1.15 Caramelos Tablero::horBarColor (Caramelos car) [protected]

Devuelve un bar horizontal del color de car.

3.21.1.16 int Tablero::movimiento (int x, int y, CaramelosMovimientos mov, Json::Value & movimientos, int maxPuntos)

Ejecuta un movimiento.

Parameters

X,:	posicion x de la ficha
у,:	posicion y de la ficha
mov,:	para donde es el movimiento
	JSON, lista de los movimientos que se realizaron.
movimientos[out],-	
:	

Returns

devuelve la cantidad de puntos que gano el usuario o 0 si no se realizo el movimiento

3.21.1.17 bool Tablero::movimientoValido (int x, int y, CaramelosMovimientos mov) [protected]

Chequea si es un movimiento valido, es decir si realizo alguna convinacion

3.21.1.18 CaramelosMovimientos Tablero::movInverso (CaramelosMovimientos *mov*) [protected]

Invierte el movimiento pasado.

Parameters

mov,:	movimiento a invertir.

Returns

movimiento invertido.

3.21.1.19 Json::Value Tablero::newMov (int x, int y, CaramelosMovimientos mov) [protected]

Crea la estructura de un movimiento para ser enviado.

Parameters

X,:	posicion en x del caramelo.
у,:	posicion en y del caramelo
mov,:	movimiento que realizo el caramelo.

Returns

la estructura de movimiento para ser enviada en formato JSON.

3.21.1.20 Json::Value Tablero::newMov (int x, int y, CaramelosMovimientos mov, Caramelos car) [protected]

Crea la estructura de un movimiento para ser enviado. Este metodo se usa para cuando aparecen caramelos

Parameters

X,:	posicion en x del caramelo.
у,:	posicion en y del caramelo
mov,:	movimiento que realizo el caramelo.
car,:	caramelo

Returns

la estructura de movimiento para ser enviada en formato JSON.

3.21.1.21 std::string Tablero::num2str (int n) [protected]

Transforma un numero a string.

Parameters

n,: numero a transformar.	
---------------------------	--

Returns

string que representa ese numero

3.21.1.22 void Tablero::rellenarTablero (Json::Value & movimientos) [protected]

Rellena los rellenar, haciendo la caida de los caramelos de mas arriba

Parameters

	movimientos de caramelos que re realizados.
movimientos[ou],-	
:	

3.21.1.23 Caramelos Tablero::verBarColor (Caramelos car) [protected]

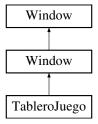
Devuelve un bar vertical del color de car.

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

- · server.tablero.h
- · server.tablero.cpp

3.22 TableroJuego Class Reference

Inheritance diagram for TableroJuego:



Public Member Functions

- TableroJuego (Json::Value mapa)
- void dibujar ()
- virtual void mensaje (Json::Value &data)

Additional Inherited Members

3.22.1 Member Function Documentation

3.22.1.1 void TableroJuego::mensaje (Json::Value & data) [virtual]

Metodo que se usa para recibir mensajes.

Parameters

```
data,: mensaje { "event": , ... }
```

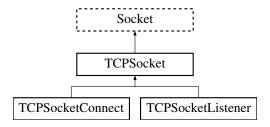
Implements Window.

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

- · cliente.tablerojuego.h
- · cliente.tablerojuego.cpp

3.23 TCPSocket Class Reference

Inheritance diagram for TCPSocket:



Additional Inherited Members

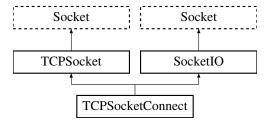
The documentation for this class was generated from the following files:

- · common.socket.h
- · common.socket.cpp

3.24 TCPSocketConnect Class Reference

#include <cliente.socket_connect.h>

Inheritance diagram for TCPSocketConnect:



Public Member Functions

- int connect (const std::string &ip)
- int connect (const int port, const std::string &ip)
- int connect (struct sockaddr_in &serv_addr)

Additional Inherited Members

3.24.1 Detailed Description

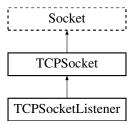
Socket TCP que permite hacer una coneccion.

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

- cliente.socket_connect.h
- · cliente.socket_connect.cpp

3.25 TCPSocketListener Class Reference

Inheritance diagram for TCPSocketListener:



Public Member Functions

- int listen (const int port)
- int listen (const int port, const std::string &ip)
- int listen (struct sockaddr_in &serv_addr)
- SocketIO * accept ()
- TCPSocketListener & setBacklog (const unsigned int backlog)

Protected Attributes

· unsigned int backlog

Additional Inherited Members

3.25.1 Member Function Documentation

3.25.1.1 SocketIO * TCPSocketListener::accept ()

Acepta una nueva conexion, devuelve un file descriptor. Bloqueante

3.25.1.2 int TCPSocketListener::listen (const int port)

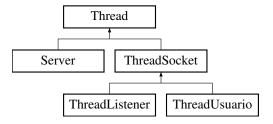
Escucha a una nueva conexion.

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

- · server.socket_listener.h
- server.socket_listener.cpp

3.26 Thread Class Reference

Inheritance diagram for Thread:



Public Member Functions

- int start ()
- int join ()
- int detach ()

Protected Member Functions

• virtual void * run ()=0

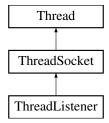
The documentation for this class was generated from the following files:

- · common.thread.h
- · common.thread.cpp

3.27 ThreadListener Class Reference

#include <cliente.thread_listener.h>

Inheritance diagram for ThreadListener:



Public Member Functions

- ThreadListener (ClienteInterface *s, SocketIO *fd)
- · void setKey (std::string &key)

Protected Member Functions

- virtual int eventNoFirmado (Json::Value &data)
- virtual int eventFirmado (Json::Value &data)
- virtual void * subRun ()

Protected Attributes

• ClienteInterface * cliente

3.27.1 Detailed Description

Thread que se encarga de leer constantemente al socket. De recibir algun mensaje lo agrega a la cola de mensajes del cliente.

3.27.2 Member Function Documentation

3.27.2.1 void ThreadListener::setKey (std::string & key)

Setear la clave usada para la firma digital de los paquetes

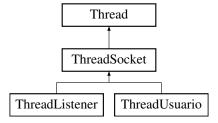
The documentation for this class was generated from the following files:

- · cliente.thread_listener.h
- · cliente.thread listener.cpp

3.28 ThreadSocket Class Reference

#include <common.thread_socket.h>

Inheritance diagram for ThreadSocket:



Public Member Functions

- ThreadSocket (SocketIO *fd)
- int **shutdown** (int how)
- int shutdown ()
- int write (Json::Value data)

Protected Member Functions

- virtual void * run ()
- virtual void * subRun ()=0
- int read (bool check=true)
- virtual int eventNoFirmado (Json::Value &data)=0
- virtual int eventFirmado (Json::Value &data)=0

Protected Attributes

- SocketIO * fd
- Mutex writeMutex
- std::string myld
- std::string key

3.28.1 Detailed Description

Thread encargado del manejo de un socket. Posee un metodo de escritura protegida con mutex, la lectura siempre se hace dentro del thread.

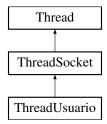
The documentation for this class was generated from the following files:

- · common.thread_socket.h
- · common.thread_socket.cpp

3.29 ThreadUsuario Class Reference

#include <server.thread_usuario.h>

Inheritance diagram for ThreadUsuario:



Public Member Functions

• ThreadUsuario (ServerInterface *s, SocketIO *fd)

Protected Member Functions

- virtual int eventNoFirmado (Json::Value &data)
- · virtual int eventFirmado (Json::Value &data)
- virtual void * subRun ()
- int onJoinGame (Json::Value &data, Json::Value &userData)
- int onLeaveGame (Json::Value &data, Json::Value &userData)
- int onNewGame (Json::Value &data, Json::Value &userData)
- int onGetMaps (Json::Value &data, Json::Value &userData)
- int welcome ()

Protected Attributes

- ServerInterface * server
- PartidaInterface * partida
- std::string user

3.29.1 Detailed Description

Thread usado para cada usuario que se conecta

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

- · server.thread_usuario.h
- · server.thread_usuario.cpp

3.30 TWavArgs Struct Reference

Public Attributes

- · int sample rate
- · int channels
- · int seconds
- int fd

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

• cliente.sound_player.cpp

3.31 TWavHeader Struct Reference

Public Attributes

- char riff [4]
- uint32_t size
- char wave [4]
- uint16_t fmt
- uint32_t length
- uint16_t type
- uint16_t channels
- uint32_t sample_rate
- uint32_t bit_rate
- uint32_t bits_per_sample
- char data [4]
- uint32_t file_size

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

cliente.sound_player.cpp

3.32 UserManager Class Reference

```
#include <common.user_manager.h>
```

Static Public Member Functions

- static void init (const std::string &path)
- static void destroy ()
- static void get (const std::string &user, Json::Value &data)
- static int set (const Json::Value &data)

Protected Member Functions

- UserManager (const std::string &path)
- void _get (const std::string &user, Json::Value &data)
- int _set (const Json::Value &data)

Protected Attributes

std::string path

· Json::Value users

Mutex mut

Static Protected Attributes

• static UserManager * me = NULL

3.32.1 Detailed Description

Clase estilo singleton encargada del manejo de toda la informacion de los usuarios. Se la debe iniciar con init y destruir con destroy.

3.32.2 Member Function Documentation

3.32.2.1 void UserManager::get (const std::string & user, Json::Value & data) [static]

Obtiene la informacion de un usario.

Parameters

user[in],:	nombre del usuario.
data[out],:	informacion del usuario

3.32.2.2 int UserManager::set (const Json::Value & data) [static]

Setea y guarda la informacion (recordar que el nombre de usaurio esta dentro de la data.

Parameters

```
data[in],: informacion del usuario
```

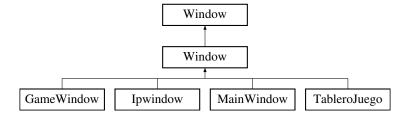
The documentation for this class was generated from the following files:

- · common.user_manager.h
- common.user_manager.cpp

3.33 Window Class Reference

#include <cliente.window.h>

Inheritance diagram for Window:



Public Types

typedef sigc::signal < void,
Json::Value > type_signal_mensaje

Public Member Functions

- virtual void mensaje (Json::Value &data)=0
- type_signal_mensaje signal_mensaje ()
- void on_about ()

Protected Member Functions

- bool _onClose (GdkEventAny *event)
- virtual bool onClose ()
- void set background image (std::string path)
- bool dialog (const char *pri, const char *sec)
- bool dialog (const std::string &pri, const std::string &sec)

Protected Attributes

• type_signal_mensaje m_signal_mensaje

3.33.1 Detailed Description

Interfaz de ventana para todas las ventanas del cliente.

3.33.2 Member Typedef Documentation

3.33.2.1 typedef sigc::signal<void, Json::Value> Window::type_signal_mensaje

Signal para avisarle al cliente que tienen que mandar un mensaje

3.33.3 Member Function Documentation

3.33.3.1 virtual void Window::mensaje (Json::Value & data) [pure virtual]

Metodo que reciben los mensajes del servidor, que deben mostrar.

Implemented in TableroJuego, Ipwindow, MainWindow, and GameWindow.

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

· cliente.window.h

Index

accept	Tablero, 25
TCPSocketListener, 30	hayMovimientos
activarCombinacionColumna	Tablero, 25
Tablero, 23	hayPatrones
activarCombinacionFila	Tablero, 25
Tablero, 23, 24	horBarColor
addUsuario	Tablero, 26
Partida, 15	,
archs	Ipwindow, 10
SoundPlayer, 21	mensaje, 11
• ,	type_signal_conectar, 11
Bar, 5	<i>7</i> – • –
broadcastMsj	listPartidas
Partida, 15	Server, 18
Button, 5	Listador, 11
	listar, 11
calcularCoordenadas	listar
Tablero, 24	Listador, 11
CandyFactory, 6	listen
Caramelo, 6	TCPSocketListener, 30
mover, 7	lock
Cliente, 7	Mutex, 14
mostrarVentanaIP, 7	log
ClienteInterface, 8	Logger, 12
nuevoMensaje, 8	Logger, 11
clientes	log, 12
Server, 18	
	MainWindow, 12
dispararColumna	mensaje, 13
Tablero, 24	mensaje
dispararFila	GameWindow, 9
Tablero, 24	Ipwindow, 11
doCombinacion	MainWindow, 13
Tablero, 24	Partida, 15
doStar	PartidaInterface, 17
Tablero, 25	TableroJuego, 28
D	Window, 36
esButton	mostrarVentanaIP
Tablero, 25	Cliente, 7
esMismoColor	movInverso
Tablero, 25	Tablero, 26
CompaNindow 0	mover
GameWindow, 8	Caramelo, 7
mensaje, 9	movimiento
on_mensaje, 9	Tablero, 26
get	movimientoValido
UserManager, 35	Tablero, 26
getNivel	
Partida, 15	msjError
havMovimiento	Partida, 16 Mutex. 13
HAVIVIOVITTIETTO	WULEX. 13

38 INDEX

lock, 14	Tablero, 22
unlock, 14	activarCombinacionColumna, 23
	activarCombinacionFila, 23, 24
newMov	calcularCoordenadas, 24
Tablero, 26, 27	dispararColumna, 24
nuevoMensaje	dispararFila, 24
ClienteInterface, 8	doCombinacion, 24
num2str	doStar, 25
Tablero, 27	esButton, 25
an manaia	esMismoColor, 25
on_mensaje	hayMovimiento, 25
GameWindow, 9	hayMovimientos, 25
Partida, 14	hayPatrones, 25
addUsuario, 15	horBarColor, 26
broadcastMsj, 15	movInverso, 26
getNivel, 15	movimiento, 26
mensaje, 15	movimientoValido, 26
msjError, 16	newMov, 26, 27
Partida, 15	num2str, 27
rmUsuario, 16	rellenarTablero, 27
PartidaInterface, 16	verBarColor, 27
mensaje, 17	TableroJuego, 28
play	mensaje, 28
SoundPlayer, 21	Thread, 30
Countriayor, 21	ThreadListener, 31
read	setKey, 32
SocketIO, 20	ThreadSocket, 32
rellenarTablero	ThreadUsuario, 33
Tablero, 27	type_signal_conectar
removePartida	Ipwindow, 11
Server, 18	type_signal_mensaje
rmUsuario	Window, 36
Partida, 16	
	unlock
Server, 17	Mutex, 14
clientes, 18	UserManager, 34
listPartidas, 18	get, 35
removePartida, 18	set, 35
ServerInterface, 18	
set	verBarColor
UserManager, 35	Tablero, 27
setKey	Window, 35
ThreadListener, 32	mensaje, 36
Socket, 19	type_signal_mensaje, 36
SocketIO, 19	write
read, 20	SocketIO, 20
write, 20	OCKETIO, 20
SoundPlayer, 20	
archs, 21	
play, 21	
Star, 21	
TCPSocket, 28	
TCPSocketConnect, 29	
TCPSocketListener, 29	
accept, 30	
listen, 30	
TWavArgs, 34	
TWavHeader, 34	