

Proyecto II - Cliente

Generated by Doxygen 1.8.14

Contents

1	File Index	1
1.1	File List	1
2	File Documentation	3
2.1	cliente.c File Reference	3
2.1.1	Detailed Description	3
2.1.2	Function Documentation	3
2.1.2.1	howto()	4
2.1.2.2	main()	4
2.1.2.3	send_error()	4
2.1.3	Variable Documentation	5
2.1.3.1	pokedex	5
2.2	cliente.h File Reference	5
2.2.1	Detailed Description	6
2.2.2	Macro Definition Documentation	6
2.2.2.1	ASK_POK	6
2.2.2.2	BAIT	7
2.2.2.3	BALL	7
2.2.2.4	BUFF_SIZE	7
2.2.2.5	ENCOUNTER	7
2.2.2.6	END_SESSION	7
2.2.2.7	ERROR_CODE	7
2.2.2.8	ESCAPE	7
2.2.2.9	MAX_POK	7

2.2.2.10	NO	8
2.2.2.11	NO_TRIES	8
2.2.2.12	NO_USER	8
2.2.2.13	OK	8
2.2.2.14	POKEDEX	8
2.2.2.15	ROCK	8
2.2.2.16	RUN	8
2.2.2.17	SERV_PORT	8
2.2.2.18	SUCCESS	9
2.2.2.19	TIMEOUT	9
2.2.2.20	TRY_USER	9
2.2.2.21	USER	9
2.2.3	Function Documentation	9
2.2.3.1	howto()	9
2.2.3.2	send_error()	9

Chapter 1

File Index

1.1 File List

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

cliente.c	Programa del Cliente	3
cliente.h	Definiciones/Mensajes del servidor y el cliente	5

Chapter 2

File Documentation

2.1 cliente.c File Reference

Programa del Cliente.

```
#include "cliente.h"
```

Functions

- int [howto](#) ()
Uso del programa.
- void [send_error](#) (int sockfd, char code)
Notifica al usuario cuando sucede algun error.
- int [main](#) (int argc, char **argv)
Función para comunicarse con el Servidor.

Variables

- const char * [pokedex](#) [[MAX_POK](#)]

2.1.1 Detailed Description

Programa del Cliente.

Date

11 Diciembre 2018 Se validan los datos del usuario, direcciones IP del servidor y el puerto, despues se empieza la conexion al servidor (si se puede claro...)

2.1.2 Function Documentation

2.1.2.1 `howto()`

```
int howto ( )
```

Uso del programa.

Returns

...

Imprime el uso del programa en caso de ser necesario

2.1.2.2 `main()`

```
int main (
    int argc,
    char ** argv )
```

Función para comunicarse con el Servidor.

Aqui se validan los datos antes de intentar una conexion al servidor, en caso de error, se manda mensaje al usuario, debe haber 4 argumentos totales, si no los hay, se manda el uso del programa. Si se tiene exito, se comienza la comunicación al servidor por medio de un menu.

Parameters

<i>argc</i>	Numero de argumentos
<i>argv</i>	Argumentos en arreglo

Returns

...

2.1.2.3 `send_error()`

```
void send_error (
    int clntfd,
    char code )
```

Notifica al usuario cuando sucede algun error.

Parameters

<i>sockfd</i>	Descriptor de Archivo
<i>code</i>	Codigo de error

Returns

Void

2.1.3 Variable Documentation

2.1.3.1 pokedex

```
const char* pokedex[MAX_POK]
```

Initial value:

```
=
{
    "Bulbasaur", "Ivysaur", "Venusaur", "Charmander", "Charmeleon", "Charizard",
    "Squirtle", "Wartortle", "Blastoise", "Caterpie", "Metapod", "Butterfree",
    "Weedle", "Kakuna", "Beedrill", "Pidgey", "Pidgeotto", "Pidgeot", "Rattata",
    "Raticate", "Spearow", "Fearow", "Ekans", "Arbok", "Pikachu", "Raichu",
    "Sandshdrew", "Sandslash", "Nidoran(F)", "Nidorina", "Nidoqueen",
    "Nidoran(M)", "Nidorino", "Nidoking", "Clefairy", "Clefable", "Vulpix",
    "Ninetales", "Jigglypuff", "Wigglytuff", "Zubat", "Golbat", "Oddish",
    "Gloom", "Vileplume", "Paras", "Parasect", "Venonat", "Venomoth", "Diglett",
    "Dugtrio", "Meowth", "Persian", "Psyduck", "Golduck", "Mankey", "Primeape",
    "Growlite", "Arcanine", "Poliwag", "Poliwhirl", "Poliwrath", "Abra",
    "Kadabra", "Alakazam", "Machop", "Machoke", "Machamp", "Bellsprout",
    "Weepinbell", "Victreebel", "Tentacool", "Tentacruel", "Geodude", "Graveler",
    "Golem", "Ponyta", "Rapidash", "Slowpoke", "Slowbro", "Magnetite",
    "Magnetron", "Farfetch'd", "Doduo", "Dodrio", "Seel", "Dewgong", "Grimer",
    "Muk", "Shellder", "Cloyster", "Gastly", "Haunter", "Gengar", "Onix",
    "Drowzee", "Hypno", "Krabby", "Klingler", "Voltorb", "Electrode", "Exeggcute",
    "Exeggutor", "Cubone", "Marowak", "Hitmonlee", "Hitmonchan", "Lickitung",
    "Koffing", "Weezing", "Rhyhorn", "Rhydon", "Chansey", "Tangela",
    "Kangaskhan", "Horsea", "Seedra", "Goldeen", "Seaking", "Staryu", "Starmie",
    "Mr. Mime", "Scyther", "Jynx", "Electabuzz", "Magmar", "Pinsir", "Tauros",
    "Magikarp", "Gyarados", "Lapras", "Ditto", "Eevee", "Vaporeon", "Jolteon",
    "Flareon", "Porygon", "Omanyte", "Omastar", "Kabuto", "Kabutops",
    "Aerodactyl", "Snorlax", "Articuno", "Zapdos", "Moltres", "Dratini",
    "Dragonair", "Dragonite", "Mewtwo", "Mew"
}
```

Pokedex Total (arreglo)

2.2 cliente.h File Reference

Definiciones/Mensajes del servidor y el cliente.

```
#include <sys/socket.h>
#include <sys/types.h>
#include <sys/time.h>
#include <netinet/in.h>
#include <netdb.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <arpa/inet.h>
```

Macros

- #define MAX_POK 151
- #define SERV_PORT 9999
- #define BUFF_SIZE 2049
- #define ASK_POK 10
- #define BALL 11
- #define ROCK 12
- #define BAIT 13
- #define RUN 14
- #define TRY_USER 15
- #define USER 20
- #define ENCOUNTER 21
- #define ESCAPE 22
- #define SUCCESS 23
- #define NO_TRIES 24
- #define POKEDEX 25
- #define OK 30
- #define NO 31
- #define END_SESSION 32
- #define TIMEOUT 40
- #define NO_USER 41
- #define ERROR_CODE 42

Functions

- int `howto` ()
Uso del programa.
- void `send_error` (int clntfd, char code)
Notifica al usuario cuando sucede algun error.

2.2.1 Detailed Description

Definiciones/Mensajes del servidor y el cliente.

Date

11 Diciembre 2018 Se define la tabla de códigos del cliente y servidor, así como de uso general y de error.

2.2.2 Macro Definition Documentation

2.2.2.1 ASK_POK

```
#define ASK_POK 10
```

(1B op) Pide por un Pokemon

2.2.2.2 BAIT

```
#define BAIT 13
```

(1B op) Tira comida al Pokemon

2.2.2.3 BALL

```
#define BALL 11
```

(1B op) Tira pokebola al Pokemon

2.2.2.4 BUFF_SIZE

```
#define BUFF_SIZE 2049
```

Tamaño de Buffer

2.2.2.5 ENCOUNTER

```
#define ENCOUNTER 21
```

(1B op, 1B id, 1B pb) Pokemon id con pb

2.2.2.6 END_SESSION

```
#define END_SESSION 32
```

(1B op) Termino de Sesion

2.2.2.7 ERROR_CODE

```
#define ERROR_CODE 42
```

(1B op) El codigo no se esperaba

2.2.2.8 ESCAPE

```
#define ESCAPE 22
```

(1B op) El Pokemon se escapo

2.2.2.9 MAX_POK

```
#define MAX_POK 151
```

Pokemon máximos

2.2.2.10 NO

```
#define NO 31
```

(1B op) Rechazar

2.2.2.11 NO_TRIES

```
#define NO_TRIES 24
```

(1B op) Ya no hay mas pokebolas

2.2.2.12 NO_USER

```
#define NO_USER 41
```

(1B op) No existe el usuario

2.2.2.13 OK

```
#define OK 30
```

(1B op) Aceptar

2.2.2.14 POKEDEX

```
#define POKEDEX 25
```

(1B op, 151B pk) Se envia a pokedex

2.2.2.15 ROCK

```
#define ROCK 12
```

(1B op) Tira piedra al Pokemon

2.2.2.16 RUN

```
#define RUN 14
```

(1B op) Escapa del encuentro

2.2.2.17 SERV_PORT

```
#define SERV_PORT 9999
```

Puerto para conexion

2.2.2.18 SUCCESS

```
#define SUCCESS 23
```

(1B op) El Pokemon se atrapo

2.2.2.19 TIMEOUT

```
#define TIMEOUT 40
```

(1B op) Hubo un timeout

2.2.2.20 TRY_USER

```
#define TRY_USER 15
```

(1B op,nB us) Conectar con el usuario us

2.2.2.21 USER

```
#define USER 20
```

(1B op) Si existe el usuario

2.2.3 Function Documentation

2.2.3.1 howto()

```
int howto ( )
```

Uso del programa.

Returns

...

Imprime el uso del programa en caso de ser necesario

2.2.3.2 send_error()

```
void send_error (
    int cIntfd,
    char code )
```

Notifica al usuario cuando sucede algun error.

Parameters

<i>sockfd</i>	Descriptor de Archivo
<i>code</i>	Codigo de error

Returns

Void

Index

ASK_POK
 cliente.h, 6

BAIT
 cliente.h, 6

BALL
 cliente.h, 7

BUFF_SIZE
 cliente.h, 7

cliente.c, 3
 howto, 3
 main, 4
 pokedex, 5
 send_error, 4

cliente.h, 5
 ASK_POK, 6
 BAIT, 6
 BALL, 7
 BUFF_SIZE, 7
 ENCOUNTER, 7
 END_SESSION, 7
 ERROR_CODE, 7
 ESCAPE, 7
 howto, 9
 MAX_POK, 7
 NO_TRIES, 8
 NO_USER, 8
 NO, 7
 OK, 8
 POKEDEX, 8
 ROCK, 8
 RUN, 8
 SERV_PORT, 8
 SUCCESS, 8
 send_error, 9
 TIMEOUT, 9
 TRY_USER, 9
 USER, 9

ENCOUNTER
 cliente.h, 7

END_SESSION
 cliente.h, 7

ERROR_CODE
 cliente.h, 7

ESCAPE
 cliente.h, 7

howto

cliente.c, 3
 cliente.h, 9

MAX_POK
 cliente.h, 7

main
 cliente.c, 4

NO_TRIES
 cliente.h, 8

NO_USER
 cliente.h, 8

NO
 cliente.h, 7

OK
 cliente.h, 8

POKEDEX
 cliente.h, 8
 pokedex
 cliente.c, 5

ROCK
 cliente.h, 8

RUN
 cliente.h, 8

SERV_PORT
 cliente.h, 8

SUCCESS
 cliente.h, 8

send_error
 cliente.c, 4
 cliente.h, 9

TIMEOUT
 cliente.h, 9

TRY_USER
 cliente.h, 9

USER
 cliente.h, 9