TouchScreenGUI

r71

Generado por Doxygen 1.5.5

Sat Jul 4 21:32:19 2009

Índice general

1. Índice de clases **1**

1.1. Lista de clases 1

2. Indice de archivos **3**

2.1. Lista de archivos 3

3. Documentación de las clases **5**

3.1. Referencia de la Clase Boton 5

3.2. Referencia de la Clase Campo 8

3.3. Referencia de la Clase Capa 11

3.4. Referencia de la Clase ClienteCapaAlta 13

3.5. Referencia de la Clase ClienteCapaBaja 15

3.6. Referencia de la Clase Dibujar 16

3.7. Referencia de la Clase DxfParser 17

3.8. Referencia de la Clase Etiqueta 19

3.9. Referencia de la Clase Frame 20

3.10. Referencia de la Clase GestorCamino 24

3.11. Referencia de la Clase GestorEstado 25

3.12. Referencia de la Clase Linea 26

3.13. Referencia de la Clase Mapa 28

3.14. Referencia de la Clase Objetivo 33

3.15. Referencia de la Clase Pantalla 37

3.16. Referencia de la Clase Polilinea 40

3.17. Referencia de la Clase Punto 42

3.18. Referencia de la Clase Radar 44

3.19. Referencia de la Clase Selector 47

3.20. Referencia de la Clase Silla 49
3.21. Referencia de la Clase Tabla 51

4. Documentación de archivos **53**

4.1. Referencia del Archivo /home/diego/proyecto/touchscreen/src/boton.cpp 53
4.2. Referencia del Archivo /home/diego/proyecto/touchscreen/src/boton.h 54
4.3. Referencia del Archivo /home/diego/proyecto/touchscreen/src/campo.cpp 55
4.4. Referencia del Archivo /home/diego/proyecto/touchscreen/src/campo.h 56
4.5. Referencia del Archivo /home/diego/proyecto/touchscreen/src/capa.cpp 57
4.6. Referencia del Archivo /home/diego/proyecto/touchscreen/src/capa.h . 58
4.7. Referencia del Archivo /home/diego/proyecto/touchscreen/src/clientecapa_-alta.cpp 59
4.8. Referencia del Archivo /home/diego/proyecto/touchscreen/src/clientecapa_-alta.h 60
4.9. Referencia del Archivo /home/diego/proyecto/touchscreen/src/clientecapa_-baja.cpp 61
4.10. Referencia del Archivo /home/diego/proyecto/touchscreen/src/clientecapa_-baja.h 62
4.11. Referencia del Archivo /home/diego/proyecto/touchscreen/src/constantes.h 63
4.12. Referencia del Archivo /home/diego/proyecto/touchscreen/src/dibujar.cpp 65
4.13. Referencia del Archivo /home/diego/proyecto/touchscreen/src/dibujar.h 66
4.14. Referencia del Archivo /home/diego/proyecto/touchscreen/src/dxfparser.cpp 67
4.15. Referencia del Archivo /home/diego/proyecto/touchscreen/src/dxfparser.h 68
4.16. Referencia del Archivo /home/diego/proyecto/touchscreen/src/etiqueta.cpp 69
4.17. Referencia del Archivo /home/diego/proyecto/touchscreen/src/etiqueta.h 70
4.18. Referencia del Archivo /home/diego/proyecto/touchscreen/src/frame.cpp 71
4.19. Referencia del Archivo /home/diego/proyecto/touchscreen/src/frame.h 72
4.20. Referencia del Archivo /home/diego/proyecto/touchscreen/src/gestorcamino.cpp 73
4.21. Referencia del Archivo /home/diego/proyecto/touchscreen/src/gestorcamino.h 74
4.22. Referencia del Archivo /home/diego/proyecto/touchscreen/src/gestorestado.cpp 75
4.23. Referencia del Archivo /home/diego/proyecto/touchscreen/src/gestorestado.h 76
4.24. Referencia del Archivo /home/diego/proyecto/touchscreen/src/gui.cpp 77
4.25. Referencia del Archivo /home/diego/proyecto/touchscreen/src/linea.cpp 79
4.26. Referencia del Archivo /home/diego/proyecto/touchscreen/src/linea.h 80
4.27. Referencia del Archivo /home/diego/proyecto/touchscreen/src/mapa.cpp 81

4.28. Referencia del Archivo /home/diego/proyecto/touchscreen/src/mapa.h 82
4.29. Referencia del Archivo /home/diego/proyecto/touchscreen/src/objetivo.cpp 83
4.30. Referencia del Archivo /home/diego/proyecto/touchscreen/src/objetivo.h 84
4.31. Referencia del Archivo /home/diego/proyecto/touchscreen/src/pantalla.cpp 85
4.32. Referencia del Archivo /home/diego/proyecto/touchscreen/src/pantalla.h 87
4.33. Referencia del Archivo /home/diego/proyecto/touchscreen/src/polilinea.cpp 88
4.34. Referencia del Archivo /home/diego/proyecto/touchscreen/src/polilinea.h 89
4.35. Referencia del Archivo /home/diego/proyecto/touchscreen/src/punto.cpp 90
4.36. Referencia del Archivo /home/diego/proyecto/touchscreen/src/punto.h 91
4.37. Referencia del Archivo /home/diego/proyecto/touchscreen/src/radar.cpp 92
4.38. Referencia del Archivo /home/diego/proyecto/touchscreen/src/radar.h 93
4.39. Referencia del Archivo /home/diego/proyecto/touchscreen/src/selector.cpp 94
4.40. Referencia del Archivo /home/diego/proyecto/touchscreen/src/selector.h 95
4.41. Referencia del Archivo /home/diego/proyecto/touchscreen/src/silla.cpp 96
4.42. Referencia del Archivo /home/diego/proyecto/touchscreen/src/silla.h . 97
4.43. Referencia del Archivo /home/diego/proyecto/touchscreen/src/tabla.cpp 98
4.44. Referencia del Archivo /home/diego/proyecto/touchscreen/src/tabla.h 99

Capítulo 1

Índice de clases

1.1. Lista de clases

Lista de las clases, estructuras, uniones e interfaces con una breve descripción:

Boton (Gestiona la entidad botón) 5
Campo (Los elementos campo almacenan información estática o numérica modificable) 8
Capa (La clase Mapa está dividida en capas) 11
ClienteCapaAlta (Gestiona la comunicación con el sistema IA. Información aperiódica) 13
ClienteCapaBaja (Gestiona la comunicacin con el sistema IA. Información periódica) 15
Dibujar (Representación gráfica del Mapa) 16
DxfParser (Parseador para el fichero DXF) 17
Etiqueta (Gestiona etiquetas de texto) 19
Frame (Gestiona el sistema de ventanas (Frames)) 20
GestorCamino (Hilo que se encarga de la lectura del camino cuando se recibe) 24
GestorEstado (Hilo que realiza la lectura de los valores de los sensores cuando se reciben) 25
Linea (Linea del mapa asignada a una capa) 26
Mapa (Organiza las capas del fichero dxf) 28
Objetivo (Clase que gestiona el objetivo establecido) 33
Pantalla (Gestiona todos los aspectos relacionados con el monitor (como la resolución)) 37
Polilinea (Elemento formado por un conjunto de Puntos) 40
Punto (Punto de coordenadas (x,y)) 42
Radar (El radar indica los obstáculos próximos y la orientación de la silla) . . 44
Selector (Gestiona la selección de mapas de un dispositivo USB) 47
Silla (Gestiona la posición y orientación de la silla) 49
Tabla (Tabla compuesta por elementos de clase Campo) 51

Capítulo 2

Indice de archivos

2.1. Lista de archivos

Lista de todos los archivos documentados y con descripciones breves:

/home/diego/proyecto/touchscreen/src/boton.cpp 53
/home/diego/proyecto/touchscreen/src/boton.h 54
/home/diego/proyecto/touchscreen/src/campo.cpp 55
/home/diego/proyecto/touchscreen/src/campo.h 56
/home/diego/proyecto/touchscreen/src/capa.cpp 57
/home/diego/proyecto/touchscreen/src/capa.h 58
/home/diego/proyecto/touchscreen/src/clientecapa_alta.cpp 59
/home/diego/proyecto/touchscreen/src/clientecapa_alta.h 60
/home/diego/proyecto/touchscreen/src/clientecapa_baja.cpp 61
/home/diego/proyecto/touchscreen/src/clientecapa_baja.h 62
/home/diego/proyecto/touchscreen/src/constantes.h 63
/home/diego/proyecto/touchscreen/src/dibujar.cpp 65
/home/diego/proyecto/touchscreen/src/dibujar.h 66
/home/diego/proyecto/touchscreen/src/dxfparser.cpp 67
/home/diego/proyecto/touchscreen/src/dxfparser.h 68
/home/diego/proyecto/touchscreen/src/etiqueta.cpp 69
/home/diego/proyecto/touchscreen/src/etiqueta.h 70
/home/diego/proyecto/touchscreen/src/frame.cpp 71
/home/diego/proyecto/touchscreen/src/frame.h 72
/home/diego/proyecto/touchscreen/src/gestorcamino.cpp 73
/home/diego/proyecto/touchscreen/src/gestorcamino.h 74
/home/diego/proyecto/touchscreen/src/gestorestado.cpp 75
/home/diego/proyecto/touchscreen/src/gestorestado.h 76
/home/diego/proyecto/touchscreen/src/gui.cpp 77
/home/diego/proyecto/touchscreen/src/linea.cpp 79
/home/diego/proyecto/touchscreen/src/linea.h 80
/home/diego/proyecto/touchscreen/src/mapa.cpp 81
/home/diego/proyecto/touchscreen/src/mapa.h 82
/home/diego/proyecto/touchscreen/src/objetivo.cpp 83

/home/diego/proyecto/touchscreen/src/objetivo.h 84
/home/diego/proyecto/touchscreen/src/pantalla.cpp 85
/home/diego/proyecto/touchscreen/src/pantalla.h 87
/home/diego/proyecto/touchscreen/src/polilinea.cpp 88
/home/diego/proyecto/touchscreen/src/polilinea.h 89
/home/diego/proyecto/touchscreen/src/punto.cpp 90
/home/diego/proyecto/touchscreen/src/punto.h 91
/home/diego/proyecto/touchscreen/src/radar.cpp 92
/home/diego/proyecto/touchscreen/src/radar.h 93
/home/diego/proyecto/touchscreen/src/selector.cpp 94
/home/diego/proyecto/touchscreen/src/selector.h 95
/home/diego/proyecto/touchscreen/src/silla.cpp 96
/home/diego/proyecto/touchscreen/src/silla.h 97
/home/diego/proyecto/touchscreen/src/tabla.cpp 98
/home/diego/proyecto/touchscreen/src/tabla.h 99

Capítulo 3

Documentación de las clases

3.1. Referencia de la Clase Boton

Gestiona la entidad botón.

`#include <boton.h>`

Métodos públicos

- Boton (SDL_Surface ∗Ventana)
- void cargarBoton (int x, int y, int w, int h, string c, Uint32 colorFondo, Uint32 colorBorde)
- void recargarBoton ()
- bool presionado (int xm, int ym)
- bool getEstado ()
- string getTexto ()
- void desactivar ()
- void deshabilitar ()
- void borrar ()
- void setIcono (string iconoruta)

3.1.1. Descripción detallada

3.1.2. Documentación del constructor y destructor

3.1.2.1. Boton::Boton (SDL_Surface ∗ *Ventana*)

Constructor, se ha de indicar donde sera pintado -> Ventana

3.1.3. Documentación de las funciones miembro

3.1.3.1. void Boton::cargarBoton (int *x*, int *y*, int *w*, int *h*, string *c*, Uint32 *colorFondo*, Uint32 *colorBorde*)

visualizar el boton en la posicion (x,y) y con una dimension de (w x h) = ancho x alto

Hace referencia a setIcono().

Referenciado por Selector::cargar(), Campo::cargarCampo(), Frame::cargarFrame(), Pantalla::entrada(), Frame::maxFrame(), Frame::minFrame(), Pantalla::minimizar(), Objetivo::preguntar(), y recargarBoton().

3.1.3.2. void Boton::recargarBoton ()

para visualizar el boton en caso de haber sido borrado

Hace referencia a cargarBoton(), y setIcono().

Referenciado por Pantalla::entrada(), Campo::handle(), y Pantalla::minimizar().

3.1.3.3. bool Boton::presionado (int *xm*, int *ym*)

devuelve true si el boton esta presionado, false en caso contrario

Referenciado por Pantalla::entrada(), y Objetivo::respuesta().

3.1.3.4. bool Boton::getEstado ()

devuelve el estado del boton, true=activo false=inactivo

3.1.3.5. string Boton::getTexto ()

devuelve el texto del boton

3.1.3.6. void Boton::desactivar ()

desactiva el boton, pero se sigue visualizando

Referenciado por Campo::cargarCampo(), Frame::desactivarFrame(), deshabilitar(), Pantalla::entrada(), y Campo::handle().

3.1.3.7. void Boton::deshabilitar ()

deshabilita el boton, oscurenciendolo

Hace referencia a desactivar().

Referenciado por Pantalla::mapaOff().

3.1.3.8. void Boton::borrar ()

borra el boton

Referenciado por Campo::cargarCampo(), y Campo::handle().

3.1.3.9. void Boton::setIcono (string *iconoruta*)

poner icono

Referenciado por cargarBoton(), Frame::cargarFrame(), Frame::maxFrame(), Frame::minFrame(), y recargarBoton().

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- /home/diego/proyecto/touchscreen/src/boton.h
- /home/diego/proyecto/touchscreen/src/boton.cpp

3.2. Referencia de la Clase Campo

Los elementos campo almacenan información estática o numérica modificable.

`#include <campo.h>`

Métodos públicos

- Campo (SDL_Surface *surface, string nombre, bool estatico, Uint32 colorNombre, Uint32 colorValor)
- void cargarCampo (int x, int y, Uint32 colorNombre, Uint32 colorValor)
- void recargar ()
- void recargar (int x, int y)
- void valorStr (string valor)
- void valorNum (float valor, float vmax, float vmin, float incremento)
- void updateValor (float valor)
- void updateValor (string valor)
- void aumentar ()
- void disminuir ()
- string getVstr ()
- bool handle (int x, int y)

3.2.1. Descripción detallada

3.2.2. Documentación del constructor y destructor

3.2.2.1. Campo::Campo (SDL_Surface * *surface*, string *nombre*, bool *estatico*, Uint32 *colorNombre*, Uint32 *colorValor*)

constructor, 'surface' donde se dibuja, 'nombre' etiqueta del campo, 'estatico' true->estatico y false->dinamico, 'colorNombre' color de la etiqueta, 'colorValor' color del valor

3.2.3. Documentación de las funciones miembro

3.2.3.1. void Campo::cargarCampo (int *x*, int *y*, Uint32 *colorNombre*, Uint32 *colorValor*)

carga el campo en la posicion 'x' 'y' con la etiqueta de color 'colorNombre' y con el valor de color 'colorValor'

Hace referencia a Boton::borrar(), Boton::cargarBoton(), y Boton::desactivar().

Referenciado por recargar().

3.2.3.2. void Campo::recargar ()

recarga el campo en su posion inicial

Hace referencia a cargarCampo().

3.2.3.3. void Campo::recargar (int *x*, int *y*)

recarga el campo en la posion 'x' 'y'

Hace referencia a cargarCampo().

3.2.3.4. void Campo::valorStr (string *valor*)

el campo tendra el valor de tipo string (en este caso no puede ser dinamico)

3.2.3.5. void Campo::valorNum (float *valor*, float *vmax*, float *vmin*, float *incremento*)

el campo tendra el valor de tipo float con un valor maximo 'vmax', un valor minimo 'vmin' y un incremento 'incremento' (solo si es dinamico)

3.2.3.6. void Campo::updateValor (float *valor*)

actuliza el valor de tipo float

Referenciado por aumentar(), y disminuir().

3.2.3.7. void Campo::updateValor (string *valor*)

actualiza el valor de tipo string

3.2.3.8. void Campo::aumentar ()

aumenta el valor (si es dinamico) una cantidad 'incremento'

Hace referencia a updateValor().

Referenciado por handle().

3.2.3.9. void Campo::disminuir ()

disminuye el valor (si es dinamico) una cantidad 'incremento'

Hace referencia a updateValor().

Referenciado por handle().

3.2.3.10. string Campo::getVstr ()

devuelve el valor actual del campo en formato string

3.2.3.11. bool Campo::handle (int *x*, int *y*)

devuelve true si se ha presionado ok

Se encarga del input mouse Retorna true si se ha presionado Ok retorna false en cualquier otro caso

Hace referencia a aumentar(), Boton::borrar(), Boton::desactivar(), disminuir(), y Boton::recargarBoton().

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- /home/diego/proyecto/touchscreen/src/campo.h
- /home/diego/proyecto/touchscreen/src/campo.cpp

3.3. Referencia de la Clase Capa

La clase Mapa está dividida en capas.

`#include <capa.h>`

Métodos públicos

- Capa (string nombre)
- vector< Linea > * getCapa ()
- vector< Polilinea > * getPolilinea ()
- void addLinea (Linea linea)
- void addPolilinea (Polilinea polilinea)
- void addVertice (Punto vertice)
- string getNombre ()
- void clear ()

3.3.1. Descripción detallada

3.3.2. Documentación del constructor y destructor

3.3.2.1. Capa::Capa (string *nombre*)

Instanciar una nueva capa con nombre 'nombre'

3.3.3. Documentación de las funciones miembro

3.3.3.1. vector< Linea > * Capa::getCapa ()

Obtener las líneas de la capa

3.3.3.2. vector< Polilinea > * Capa::getPolilinea ()

Obtener las polilíneas de la capa

Referenciado por DxfParser::addVertex(), y Pantalla::entrada().

3.3.3.3. void Capa::addLinea (Linea *linea*)

Añadir una lí a la capa

Referenciado por DxfParser::addVertex().

3.3.3.4. void Capa::addPolilinea (Polilinea *polilinea*)

Añadir una polilí a la capa

3.3.3.5. void Capa::addVertice (Punto *vertice*)

Añadir un vértice a la capa

Referenciado por DxfParser::addVertex().

3.3.3.6. string Capa::getNombre ()

Obtener el nombre de la capa

3.3.3.7. void Capa::clear ()

Limpiar el vector de polilíneas

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- /home/diego/proyecto/touchscreen/src/capa.h
- /home/diego/proyecto/touchscreen/src/capa.cpp

3.4. Referencia de la Clase ClienteCapaAlta

Gestiona la comunicación con el sistema IA. Información aperiódica.

`#include <clientecapa_alta.h>`

Métodos públicos

- void setMap (string path)
- vector< Punto > getCamino ()
- void clearCamino ()
- void enviarPlano (string path)
- void enviar (string dato)
- void conectar ()

3.4.1. Descripción detallada

3.4.2. Documentación de las funciones miembro

3.4.2.1. void ClienteCapaAlta::setMap (string *path*)

Carga el mapa situado en "path"

3.4.2.2. vector< Punto > ClienteCapaAlta::getCamino ()

Obtiene el camino guardado

3.4.2.3. void ClienteCapaAlta::clearCamino ()

Borra el camino guardado

3.4.2.4. void ClienteCapaAlta::enviarPlano (string *path*)

Envia el plano situado en "path" al sistema IA

Referenciado por Pantalla::entrada().

3.4.2.5. void ClienteCapaAlta::enviar (string *dato*)

Envia la información "dato" al sistema IA

Referenciado por Pantalla::entrada().

3.4.2.6. void ClienteCapaAlta::conectar ()

Conecta con el sistema IA

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- /home/diego/proyecto/touchscreen/src/clientecapa_alta.h
- /home/diego/proyecto/touchscreen/src/clientecapa_alta.cpp

3.5. Referencia de la Clase ClienteCapaBaja

Gestiona la comunicacin con el sistema IA. Información periódica.

`#include <clientecapa_baja.h>`

Métodos públicos

- vector< double > getValores ()
- void conectar ()

3.5.1. Descripción detallada

3.5.2. Documentación de las funciones miembro

3.5.2.1. vector< double > ClienteCapaBaja::getValores ()

devuelve los valores obtenidos

3.5.2.2. void ClienteCapaBaja::conectar ()

Conectar con el servidor IA

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- /home/diego/proyecto/touchscreen/src/clientecapa_baja.h
- /home/diego/proyecto/touchscreen/src/clientecapa_baja.cpp

3.6. Referencia de la Clase Dibujar

Representación gráfica del Mapa.

`#include <dibujar.h>`

Métodos públicos

- Dibujar (SDL_Surface *screen)
- void dibujarLinea (Frame *frame, Linea *linea, Uint32 color)

3.6.1. Descripción detallada

3.6.2. Documentación del constructor y destructor

3.6.2.1. Dibujar::Dibujar (SDL_Surface * *screen*)

Instancia para dibujar en 'screen'

3.6.3. Documentación de las funciones miembro

3.6.3.1. void Dibujar::dibujarLinea (Frame * *frame*, Linea * *linea*, Uint32 *color*)

Dibujar una linea 'linea' de color 'color en el frame 'frame'

Hace referencia a Frame::activarFrame(), Linea::getX1(), Linea::getX2(), Linea::getY1(), y Linea::getY2().

Referenciado por Mapa::pintarCamino(), y Mapa::pintarMapa().

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- /home/diego/proyecto/touchscreen/src/dibujar.h
- /home/diego/proyecto/touchscreen/src/dibujar.cpp

3.7. Referencia de la Clase DxfParser

Parseador para el fichero DXF.

`#include <dxfparser.h>`

Métodos públicos

- DxfParser ()
- virtual void addLayer (const DL_LayerData &data)
- virtual void addLine (const DL_LineData &data)
- virtual void addPolyline (const DL_PolylineData &data)
- virtual void addVertex (const DL_VertexData &data)
- virtual void addBlock (const DL_BlockData &data)

Atributos públicos

- string nombre

3.7.1. Descripción detallada

3.7.2. Documentación del constructor y destructor

3.7.2.1. DxfParser::DxfParser ()

constructor

3.7.3. Documentación de las funciones miembro

3.7.3.1. void DxfParser::addLayer (const DL_LayerData & *data*) [virtual]

Añade una nueva Capa a la estructura Mapa

Hace referencia a Mapa::addCapa().

3.7.3.2. void DxfParser::addLine (const DL_LineData & *data*) [virtual]

Añade una nueva Linea a la estructura Mapa

Hace referencia a Mapa::getCapa().

3.7.3.3. void DxfParser::addPolyline (const DL_PolylineData & *data*) [virtual]

Añade una nueva Polilinea a la estructura Mapa

Hace referencia a Mapa::getCapa(), y nombre.

3.7.3.4. void DxfParser::addVertex (const DL_VertexData & *data*) [virtual]

Añade un nuevo Punto a la estructura Mapa

Hace referencia a Capa::addLinea(), Capa::addVertice(), Mapa::getCapa(), y Capa::getPolilinea().

3.7.3.5. void DxfParser::addBlock (const DL_BlockData & *data*) [virtual]

Añade un nuevo Bloque a la estructura Mapa

Hace referencia a nombre.

3.7.4. Documentación de los datos miembro

3.7.4.1. string DxfParser::nombre

variable temporal para guardar el nombre del bloque

Referenciado por addBlock(), y addPolyline().

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- /home/diego/proyecto/touchscreen/src/dxfparser.h
- /home/diego/proyecto/touchscreen/src/dxfparser.cpp

3.8. Referencia de la Clase Etiqueta

Gestiona etiquetas de texto.

`#include <etiqueta.h>`

Métodos públicos

- Etiqueta (SDL_Surface ∗Ventana)
- void cargarEtiqueta (int x, int y, int w, int h, string c, Uint32 colorFuente, Uint32 colorBorde, Uint32 colorRelleno)
- void dibujarEtiqueta ()
- void insertarTexto (string c)

3.8.1. Descripción detallada

3.8.2. Documentación del constructor y destructor

3.8.2.1. Etiqueta::Etiqueta (SDL_Surface ∗ *Ventana*)

Especificar la "Ventana" donde se situa

3.8.3. Documentación de las funciones miembro

3.8.3.1. void Etiqueta::cargarEtiqueta (int *x*, int *y*, int *w*, int *h*, string *c*, Uint32 *colorFuente*, Uint32 *colorBorde*, Uint32 *colorRelleno*)

Carga la etiqueta en la posicion 'x,y' con una anchura 'w' y una altura 'h'. El texto 'c'

Referenciado por Frame::cargarFrame(), Pantalla::entrada(), Frame::maxFrame(), Frame::minFrame(), y Pantalla::minimizar().

3.8.3.2. void Etiqueta::dibujarEtiqueta ()

Dibuja la etiqueta con la configuración inicial

3.8.3.3. void Etiqueta::insertarTexto (string *c*)

Inserta el texto 'c' en la etiqueta

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- /home/diego/proyecto/touchscreen/src/etiqueta.h
- /home/diego/proyecto/touchscreen/src/etiqueta.cpp

3.9. Referencia de la Clase Frame

Gestiona el sistema de ventanas (Frames).

`#include <frame.h>`

Métodos públicos

- Frame (SDL_Surface *ventana)
- void cargarFrame (int x, int y, int w, int h, string c, Uint32 color)
- void maxFrame (int x, int y, int w, int h)
- void minFrame ()
- void desactivarFrame ()
- void limpiarFrame (bool refresh)
- void activarFrame ()
- void refrescarFrame ()
- bool presionado (int xm, int ym)
- SDL_Surface * getVentana ()
- int getX ()
- int getY ()
- int getW ()
- int getH ()
- SDL_Rect getArea ()
- Boton * getBmaxmin ()
- Uint8 getEstado ()

3.9.1. Descripción detallada

3.9.2. Documentación del constructor y destructor

3.9.2.1. Frame::Frame (SDL_Surface * *ventana*)

Instancia de frame en la ventana 'ventana'

3.9.3. Documentación de las funciones miembro

3.9.3.1. void Frame::cargarFrame (int *x*, int *y*, int *w*, int *h*, string *c*, Uint32 *color*)

Carga el frame en la posicion 'x' 'y' con una anchura 'w' y una altura 'h'. El título de la ventana es 'c', y el borde de color 'color'

Hace referencia a Boton::cargarBoton(), Etiqueta::cargarEtiqueta(), limpiarFrame(), y Boton::setIcono().

Referenciado por Selector::cargar().

3.9.3.2. void Frame::maxFrame (int *x*, int *y*, int *w*, int *h*)

Maximiza el frame

Hace referencia a Boton::cargarBoton(), Etiqueta::cargarEtiqueta(), limpiarFrame(), y Boton::setIcono().

Referenciado por Pantalla::entrada().

3.9.3.3. void Frame::minFrame ()

Minimiza el frame

Hace referencia a Boton::cargarBoton(), Etiqueta::cargarEtiqueta(), getX(), limpiarFrame(), y Boton::setIcono().

Referenciado por Pantalla::minimizar().

3.9.3.4. void Frame::desactivarFrame ()

Desactiva el frame

Hace referencia a Boton::desactivar().

Referenciado por Pantalla::entrada(), Pantalla::mapaOff(), y Pantalla::minimizar().

3.9.3.5. void Frame::limpiarFrame (bool *refresh*)

Limpia el frame. Si refresh=true se refresca al momento.

Hace referencia a getArea().

Referenciado por cargarFrame(), Pantalla::entrada(), maxFrame(), minFrame(), y Mapa::pintarMapa().

3.9.3.6. void Frame::activarFrame ()

Activa el frame

Hace referencia a getArea().

Referenciado por Dibujar::dibujarLinea(), Radar::Radar(), y Radar::recargar().

3.9.3.7. void Frame::refrescarFrame ()

Refresca la zona del frame

Hace referencia a getArea().

Referenciado por Silla::dibujar(), Objetivo::dibujar(), Pantalla::entrada(), y Mapa::pintarCamino().

3.9.3.8. bool Frame::presionado (int *xm*, int *ym*)

Devuelve true si se ha presionado el mouse dentro del frame

Hace referencia a getH(), getW(), getX(), y getY().

Referenciado por Pantalla::entrada().

3.9.3.9. SDL_Surface * Frame::getVentana ()

Devuelve la Surface que contiene al frame

Referenciado por Mapa::despAbajo(), Mapa::despArriba(), Mapa::despDerecha(), Mapa::despIzquierda(), Silla::dibujar(), Objetivo::dibujar(), Mapa::escalarMapa(), Radar::Radar(), Radar::recargar(), y Objetivo::setObjetivo().

3.9.3.10. int Frame::getX ()

Devuelve la posició X del frame

Referenciado por Selector::cargar(), Punto::cpantalla(), Punto::cplano(), Pantalla::entrada(), minFrame(), Pantalla::minimizar(), Objetivo::preguntar(), presionado(), Radar::Radar(), Tabla::recargar(), Radar::recargar(), y Pantalla::setAlpha().

3.9.3.11. int Frame::getY ()

Devuelve la posició Y del frame

Referenciado por Selector::cargar(), Punto::cpantalla(), Punto::cplano(), Pantalla::entrada(), Pantalla::minimizar(), Objetivo::preguntar(), presionado(), Radar::Radar(), Tabla::recargar(), Radar::recargar(), y Pantalla::setAlpha().

3.9.3.12. int Frame::getW ()

Devuelve la anchura del frame

Referenciado por Mapa::calcularDHV(), Selector::cargar(), Pantalla::entrada(), Pantalla::minimizar(), Objetivo::preguntar(), presionado(), Radar::Radar(), Radar::recargar(), y Pantalla::setAlpha().

3.9.3.13. int Frame::getH ()

Devuelve la altura del frame

Referenciado por Mapa::calcularDHV(), Punto::cpantalla(), Punto::cplano(), Pantalla::entrada(), Pantalla::minimizar(), Objetivo::preguntar(), presionado(), Radar::Radar(), Tabla::recargar(), Radar::recargar(), y Pantalla::setAlpha().

3.9.3.14. SDL_Rect Frame::getArea ()

Devuelve el area del grame

Referenciado por activarFrame(), Silla::dibujar(), Objetivo::dibujar(), limpiarFrame(), y refrescarFrame().

3.9.3.15. Boton * Frame::getBmaxmin ()

Devuelve el bot&oacute del frame

Referenciado por Pantalla::entrada().

3.9.3.16. Uint8 Frame::getEstado ()

Devuelve el estado del frame

Referenciado por Pantalla::entrada(), y Tabla::recargar().

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- /home/diego/proyecto/touchscreen/src/frame.h
- /home/diego/proyecto/touchscreen/src/frame.cpp

3.10. Referencia de la Clase GestorCamino

Hilo que se encarga de la lectura del camino cuando se recibe.

`#include <gestorcamino.h>`

Métodos públicos

- GestorCamino (SDL_Surface *surface)

3.10.1. Descripción detallada

3.10.2. Documentación del constructor y destructor

3.10.2.1. GestorCamino::GestorCamino (SDL_Surface * *surface*)

Instancia del gestor de caminos

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- /home/diego/proyecto/touchscreen/src/gestorcamino.h
- /home/diego/proyecto/touchscreen/src/gestorcamino.cpp

3.11. Referencia de la Clase GestorEstado

Hilo que realiza la lectura de los valores de los sensores cuando se reciben.

`#include <gestorestado.h>`

3.11.1. Descripción detallada

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- /home/diego/proyecto/touchscreen/src/gestorestado.h
- /home/diego/proyecto/touchscreen/src/gestorestado.cpp

3.12. Referencia de la Clase Linea

linea del mapa asignada a una capa

`#include <linea.h>`

Métodos públicos

- Linea (string capa, double x1, double y1, double x2, double y2)
- Linea (string capa, Punto v1, Punto v2)
- double getX1 ()
- double getY1 ()
- double getX2 ()
- double getY2 ()
- string getCapa ()

3.12.1. Descripción detallada

3.12.2. Documentación del constructor y destructor

3.12.2.1. Linea::Linea (string *capa*, double *x1*, double *y1*, double *x2*, double *y2*)

Crear una linea en la capa 'capa', de origen (x1,y1) y destino (x2,y2)

3.12.2.2. Linea::Linea (string *capa*, Punto *v1*, Punto *v2*)

Crear una linea en la capa 'capa' de origen v1 y destino v2

Hace referencia a Punto::getX(), y Punto::getY().

3.12.3. Documentación de las funciones miembro

3.12.3.1. double Linea::getX1 ()

Devuelve x1 de la linea

Referenciado por Mapa::calcularZoom(), y Dibujar::dibujarLinea().

3.12.3.2. double Linea::getY1 ()

Devuelve y1 de la linea

Referenciado por Mapa::calcularZoom(), y Dibujar::dibujarLinea().

3.12.3.3. double Linea::getX2 ()

Devuelve x2 de la linea

Referenciado por Mapa::calcularZoom(), y Dibujar::dibujarLinea().

3.12.3.4. double Linea::getY2 ()

Devuelve y2 de la linea

Referenciado por Mapa::calcularZoom(), y Dibujar::dibujarLinea().

3.12.3.5. string Linea::getCapa ()

Devuelve el nombre de la capa a la que pertenece la línea

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- /home/diego/proyecto/touchscreen/src/linea.h
- /home/diego/proyecto/touchscreen/src/linea.cpp

3.13. Referencia de la Clase Mapa

Organiza las capas del fichero dxf.

`#include <mapa.h>`

Métodos públicos

- Mapa ()
- string getPath ()
- void lectura (string ruta)
- double getDH ()
- double getDV ()
- double getEscala ()
- char * getEscalaStr ()
- vector< Capa > * getMapa ()
- Capa * getCapa (string c)
- void addCapa (Capa capa)
- void setCamino (vector< Linea > c)
- void clearCamino ()
- void clearMapa ()
- void escalarMapa (double escala)
- void pintarMapa (SDL_Surface *screen, double escala)
- void pintarCamino (SDL_Surface *screen, Frame *frame, double escala)
- void calcularDHV (Frame *frame)
- void centrarMapa ()
- void calcularZoom ()
- void setFrame (Frame *frame)
- void despArriba ()
- void despAbajo ()
- void despIzquierda ()
- void despDerecha ()

3.13.1. Descripción detallada

3.13.2. Documentación del constructor y destructor

3.13.2.1. Mapa::Mapa ()

Instancia del mapa

3.13.3. Documentación de las funciones miembro

3.13.3.1. string Mapa::getPath ()

Devuelve la ruta del mapa actual

Referenciado por Pantalla::entrada().

3.13.3.2. void Mapa::lectura (string *ruta*)

Lectura de un mapa situado en 'ruta'

Referenciado por Selector::handle().

3.13.3.3. double Mapa::getDH ()

Duelve el desplazamiento horizontal del origen

Referenciado por Silla::dibujar(), Objetivo::dibujar(), Pantalla::entrada(), y Objetivo::preguntar().

3.13.3.4. double Mapa::getDV ()

Duelve el desplazamiento vertical del origen

Referenciado por Silla::dibujar(), Objetivo::dibujar(), Pantalla::entrada(), y Objetivo::preguntar().

3.13.3.5. double Mapa::getEscala ()

Duelve la escala del mapa en formato numérico

Referenciado por Silla::dibujar(), Objetivo::dibujar(), Pantalla::entrada(), Pantalla::minimizar(), y Objetivo::preguntar().

3.13.3.6. char * Mapa::getEscalaStr ()

Duelve la escala del mapa en formato cadena

3.13.3.7. vector< Capa > * Mapa::getMapa ()

Devuelve las capas que contiene el mapa

3.13.3.8. Capa * Mapa::getCapa (string *c*)

Devuelve la capa de nombre 'c'

Referenciado por DxfParser::addLine(), DxfParser::addPolyline(), DxfParser::addVertex(), y Pantalla::entrada().

3.13.3.9. void Mapa::addCapa (Capa *capa*)

Añ la capa 'capa' al mapa

Referenciado por DxfParser::addLayer().

3.13.3.10. void Mapa::setCamino (vector< Linea > *c*)

El camino al objetivo pasa a ser 'c'

3.13.3.11. void Mapa::clearCamino ()

Borrar camino

3.13.3.12. void Mapa::clearMapa ()

Borra las capas del mapa

3.13.3.13. void Mapa::escalarMapa (double *escala*)

Escala el mapa a 'escala'

Hace referencia a Frame::getVentana(), y pintarMapa().

Referenciado por Pantalla::entrada().

3.13.3.14. void Mapa::pintarMapa (SDL_Surface * *screen*, double *escala*)

Pinta el mapa en la superficie 'screen', en el frame 'frame', a escala 'escala'

Hace referencia a calcularDHV(), Dibujar::dibujarLinea(), Frame::limpiarFrame(), y pintarCamino().

Referenciado por despAbajo(), despArriba(), despDerecha(), despIzquierda(), Silla::dibujar(), Pantalla::entrada(), escalarMapa(), y Pantalla::minimizar().

3.13.3.15. void Mapa::pintarCamino (SDL_Surface * *screen*, Frame * *frame*, double *escala*)

Pinta el camino en la superficie 'screen', en el frame 'frame', a escala 'escala'

Hace referencia a Dibujar::dibujarLinea(), y Frame::refrescarFrame().

Referenciado por pintarMapa().

3.13.3.16. void Mapa::calcularDHV (Frame * *frame*)

Calculo de los desplazamientos para el centrado

Hace referencia a Frame::getH(), y Frame::getW().

Referenciado por calcularZoom(), y pintarMapa().

3.13.3.17. void Mapa::centrarMapa ()

Centra el mapa

Referenciado por Pantalla::entrada(), y Pantalla::minimizar().

3.13.3.18. void Mapa::calcularZoom ()

Calcular el zoom para ajustar el mapa a la ventana

Hace referencia a calcularDHV(), Linea::getX1(), Linea::getX2(), Linea::getY1(), y Linea::getY2().

Referenciado por Pantalla::entrada(), y Pantalla::minimizar().

3.13.3.19. void Mapa::setFrame (Frame * *frame*)

Frame donde se pintara

3.13.3.20. void Mapa::despArriba ()

Desplaza el mapa arriba

Hace referencia a Frame::getVentana(), y pintarMapa().

Referenciado por Pantalla::entrada().

3.13.3.21. void Mapa::despAbajo ()

Desplaza el mapa abajo

Hace referencia a Frame::getVentana(), y pintarMapa().

Referenciado por Pantalla::entrada().

3.13.3.22. void Mapa::despIzquierda ()

Desplaza el mapa a la izquierda

Hace referencia a Frame::getVentana(), y pintarMapa().

Referenciado por Pantalla::entrada().

3.13.3.23. void Mapa::despDerecha ()

Desplaza el mapa a la derecha

Hace referencia a Frame::getVentana(), y pintarMapa().

Referenciado por Pantalla::entrada().

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- /home/diego/proyecto/touchscreen/src/mapa.h
- /home/diego/proyecto/touchscreen/src/mapa.cpp

3.14. Referencia de la Clase Objetivo

Clase que gestiona el objetivo establecido.

`#include <objetivo.h>`

Métodos públicos

- Objetivo ()
- Objetivo (Frame *frame, Mapa *plano, double xp, double yp)
- void setObjetivo (Frame *frame, Mapa *plano, double xp, double yp)
- bool interior (Polilinea polilinea)
- void dibujar ()
- void activar ()
- void desactivar ()
- bool getFijado ()
- void setValido (bool valido)
- bool getValido ()
- double getX ()
- double getY ()
- void preguntar ()
- int respuesta (int x, int y)
- bool preguntado ()
- void nopreguntar ()
- void load ()
- void store ()
- string toString ()

3.14.1. Descripción detallada

3.14.2. Documentación del constructor y destructor

3.14.2.1. Objetivo::Objetivo ()

Instancia del objetivo

3.14.2.2. Objetivo::Objetivo (Frame * *frame*, Mapa * *plano*, double *xp*, double *yp*)

Instancia del objetivo en el plano 'plano' en el punto (xp,yp)

Hace referencia a setObjetivo().

3.14.3. Documentación de las funciones miembro

3.14.3.1. void Objetivo::setObjetivo (Frame * *frame*, Mapa * *plano*, double *xp*, double *yp*)

El objetivo en el plano 'plano' en el punto (xp,yp)

Hace referencia a Frame::getVentana().

Referenciado por Pantalla::entrada(), y Objetivo().

3.14.3.2. bool Objetivo::interior (Polilinea *polilinea*)

Test del rayo en 2D sobre el poligono 'polilinea'

Hace referencia a Polilinea::toLineas().

Referenciado por Pantalla::entrada().

3.14.3.3. void Objetivo::dibujar ()

Dibujar el objetivo

Hace referencia a Punto::cpantalla(), Frame::getArea(), Mapa::getDH(), Mapa::getDV(), Mapa::getEscala(), Frame::getVentana(), Punto::getX(), Punto::getY(), y Frame::refrescarFrame().

Referenciado por Pantalla::entrada(), y Pantalla::minimizar().

3.14.3.4. void Objetivo::activar ()

Activa el objetivo

Referenciado por Pantalla::entrada().

3.14.3.5. void Objetivo::desactivar ()

Desactiva el objetivo

3.14.3.6. bool Objetivo::getFijado ()

Devuelve si el objetivo está fijado o no

Referenciado por Pantalla::entrada(), y Pantalla::minimizar().

3.14.3.7. void Objetivo::setValido (bool *valido*)

Si es vá o no

Referenciado por Pantalla::entrada().

3.14.3.8. bool Objetivo::getValido ()

Si es inapropiado

Referenciado por Pantalla::entrada().

3.14.3.9. double Objetivo::getX ()

Devuelve la componente X del objetivo

Referenciado por preguntar().

3.14.3.10. double Objetivo::getY ()

Devuelve la componente Y del objetivo

Referenciado por preguntar().

3.14.3.11. void Objetivo::preguntar ()

Pregunta si se ha de fijar el objetivo

Hace referencia a Boton::cargarBoton(), Punto::cpantalla(), Mapa::getDH(), Mapa::getDV(), Mapa::getEscala(), Frame::getH(), Frame::getW(), getX(), Frame::getX(), Punto::getX(), Frame::getY(), getY(), y Punto::getY().

Referenciado por Pantalla::entrada().

3.14.3.12. int Objetivo::respuesta (int *x*, int *y*)

Devuelve la respuesta dada por el usuario

Hace referencia a Boton::presionado().

Referenciado por Pantalla::entrada().

3.14.3.13. bool Objetivo::preguntado ()

Devulve si se ha realizado la pregunta

Referenciado por Pantalla::entrada(), y Pantalla::minimizar().

3.14.3.14. void Objetivo::nopreguntar ()

Se deja de preguntar

Referenciado por Pantalla::entrada(), y Pantalla::minimizar().

3.14.3.15. void Objetivo::load ()

Carga el objetivo guardado temporalmente

Referenciado por Pantalla::entrada(), y Pantalla::minimizar().

3.14.3.16. void Objetivo::store ()

Guarda el objetivo temporalmente

Referenciado por Pantalla::entrada().

3.14.3.17. string Objetivo::toString ()

Devuelve el objetivo en formato cadena

Referenciado por Pantalla::entrada().

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- /home/diego/proyecto/touchscreen/src/objetivo.h
- /home/diego/proyecto/touchscreen/src/objetivo.cpp

3.15. Referencia de la Clase Pantalla

Gestiona todos los aspectos relacionados con el monitor (como la resolución).

`#include <pantalla.h>`

Métodos públicos

- Pantalla (SDL_Surface *screen)
- SDL_Surface * getPantalla ()
- void hideCursor ()
- void entrada ()
- void borrar ()
- void setAlpha (Frame *frame, Uint8 zona)
- bool salir ()
- void minimizar ()
- void mapaOff ()
- void setHandle (bool handle)

3.15.1. Descripción detallada

3.15.2. Documentación del constructor y destructor

3.15.2.1. Pantalla::Pantalla (SDL_Surface * *screen*)

Instancia de la pantalla

3.15.3. Documentación de las funciones miembro

3.15.3.1. SDL_Surface * Pantalla::getPantalla ()

Devuelve el puntero a la pantalla

3.15.3.2. void Pantalla::hideCursor ()

Ocultar el cursor

3.15.3.3. void Pantalla::entrada ()

Gestiona la entrada del teclado

Hace referencia a Objetivo::activar(), borrar(), Selector::buscarW(), Mapa::calcularZoom(), Boton::cargarBoton(), Etiqueta::cargarEtiqueta(), Mapa::centrarMapa(), Punto::cplano(), Boton::desactivar(), Frame::desactivarFrame(), Mapa::despAbajo(), Mapa::despArriba(), Mapa::despDerecha(), Mapa::despIzquierda(), Silla::dibujar(), Objetivo::dibujar(), ClienteCapaAlta::enviar(), ClienteCapaAlta::enviarPlano(), Mapa::escalarMapa(), Frame::getBmaxmin(), Mapa::getCapa(), Mapa::getDH(), Mapa::getDV(), Mapa::getEscala(), Frame::getEstado(), Objetivo::getFijado(), Frame::getH(), Mapa::getPath(), Capa::getPolilinea(), Silla::getStatus(), Objetivo::getValido(), Frame::getW(), Frame::getX(), Punto::getX(), Frame::getY(), Punto::getY(), Tabla::handle(), Selector::handle(), Objetivo::interior(), Frame::limpiarFrame(), Objetivo::load(), Frame::maxFrame(), minimizar(), Objetivo::nopreguntar(), Mapa::pintarMapa(), Objetivo::preguntado(), Objetivo::preguntar(), Boton::presionado(), Frame::presionado(), Tabla::recargar(), Radar::recargar(), Boton::recargarBoton(), Frame::refrescarFrame(), Objetivo::respuesta(), Objetivo::setObjetivo(), Objetivo::setValido(), Objetivo::store(), Silla::toogleStatus(), y Objetivo::toString().

3.15.3.4. void Pantalla::borrar ()

Limpia la pantalla

Referenciado por entrada().

3.15.3.5. void Pantalla::setAlpha (Frame * *frame*, Uint8 *zona*)

Oscurece la zona indicada

Hace referencia a Frame::getH(), Frame::getW(), Frame::getX(), y Frame::getY().

3.15.3.6. bool Pantalla::salir ()

Devuelve si se ha salido del programa

3.15.3.7. void Pantalla::minimizar ()

Minimiza las 3 ventanas

Hace referencia a Mapa::calcularZoom(), Boton::cargarBoton(), Etiqueta::cargarEtiqueta(), Mapa::centrarMapa(), Frame::desactivarFrame(), Objetivo::dibujar(), Silla::dibujar(), Mapa::getEscala(), Objetivo::getFijado(), Frame::getH(), Frame::getW(), Frame::getX(), Frame::getY(), Objetivo::load(), Frame::minFrame(), Objetivo::nopreguntar(), Mapa::pintarMapa(), Objetivo::preguntado(), Tabla::recargar(), y Boton::recargarBoton().

Referenciado por entrada(), y Selector::handle().

3.15.3.8. void Pantalla::mapaOff ()

Desactiva el mapa

Hace referencia a Frame::desactivarFrame(), y Boton::deshabilitar().

Referenciado por Selector::cargar().

3.15.3.9. void Pantalla::setHandle (bool *handle*)

Modifica el valor de gestionar o no la entrada

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- /home/diego/proyecto/touchscreen/src/pantalla.h
- /home/diego/proyecto/touchscreen/src/pantalla.cpp

3.16. Referencia de la Clase Polilinea

Elemento formado por un conjunto de Puntos.

`#include <polilinea.h>`

Métodos públicos

- Polilinea (int num, bool cerrado, string capa)
- vector< Punto > * getPolilinea ()
- void addVertice (Punto vertice)
- bool getCerrado ()
- int getNumTotal ()
- int getNum ()
- vector< Linea > toLineas ()
- void setHabitacion (string nombre)
- string getHabitacion ()

3.16.1. Descripción detallada

3.16.2. Documentación del constructor y destructor

3.16.2.1. Polilinea::Polilinea (int *num*, bool *cerrado*, string *capa*)

Instancia de una polilínea en la capa 'capa', con 'num' vértices, 'cerrado' si es cerrado o abierto

3.16.3. Documentación de las funciones miembro

3.16.3.1. vector< Punto > * Polilinea::getPolilinea ()

Devuelve la lista de puntos de la polilínea

3.16.3.2. void Polilinea::addVertice (Punto *vertice*)

Añadir un vé a la polilínea

3.16.3.3. bool Polilinea::getCerrado ()

Devuelve si la polilínea es cerrada

3.16.3.4. int Polilinea::getNumTotal ()

Devuelve el número total de vértices de la polilínea

3.16.3.5. int Polilinea::getNum ()

Devuelve el número de vértices de la polilínea

3.16.3.6. vector< Linea > Polilinea::toLineas ()

Transforma la lista de puntos de la polilínea en líneas

Referenciado por Objetivo::interior().

3.16.3.7. void Polilinea::setHabitacion (string *nombre*)

Poner nombre a la polilínea como habitación

3.16.3.8. string Polilinea::getHabitacion ()

Devuelve el nombre de la habitación

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- /home/diego/proyecto/touchscreen/src/polilinea.h
- /home/diego/proyecto/touchscreen/src/polilinea.cpp

3.17. Referencia de la Clase Punto

Punto de coordenadas (x,y).

`#include <punto.h>`

Métodos públicos

- Punto ()
- Punto (double x, double y)
- double getX ()
- double getY ()
- void cpantalla (Frame *frame, double dh, double dv, double escala)
- void cplano (double x, double y, Frame *frame, double dh, double dv, double escala)

3.17.1. Descripción detallada

3.17.2. Documentación del constructor y destructor

3.17.2.1. Punto::Punto ()

Instancia de un punto

3.17.2.2. Punto::Punto (double *x*, double *y*)

Instancia de un punto (x,y)

3.17.3. Documentación de las funciones miembro

3.17.3.1. double Punto::getX ()

Devuelve la componente X del punto

Referenciado por Silla::dibujar(), Objetivo::dibujar(), Pantalla::entrada(), Linea::Linea(), y Objetivo::preguntar().

3.17.3.2. double Punto::getY ()

Devuelve la componente Y del punto

Referenciado por Silla::dibujar(), Objetivo::dibujar(), Pantalla::entrada(), Linea::Linea(), y Objetivo::preguntar().

3.17.3.3. void Punto::cpantalla (Frame * *frame*, double *dh*, double *dv*, double *escala*)

Cambiar sistema cordenadas de plano a pantalla

Hace referencia a Frame::getH(), Frame::getX(), y Frame::getY().

Referenciado por Silla::dibujar(), Objetivo::dibujar(), y Objetivo::preguntar().

3.17.3.4. void Punto::cplano (double *x*, double *y*, Frame * *frame*, double *dh*, double *dv*, double *escala*)

Cambiar sistema cordenadas de pantalla a plano

Hace referencia a Frame::getH(), Frame::getX(), y Frame::getY().

Referenciado por Pantalla::entrada().

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- /home/diego/proyecto/touchscreen/src/punto.h
- /home/diego/proyecto/touchscreen/src/punto.cpp

3.18. Referencia de la Clase Radar

El radar indica los obstáculos próximos y la orientación de la silla.

`#include <radar.h>`

Métodos públicos

- Radar (Frame *frame)
- void recargar (bool refresh)
- int getX ()
- int getY ()
- int getR1 ()
- int getR2 ()
- int getR3 ()
- Frame * getFrame ()
- vector< Punto > getObstaculos ()
- void addObstaculo (Punto o)
- void dibujarFlecha (int rot)
- SDL_Surface * getFlecha ()
- SDL_Rect getDesp ()

3.18.1. Descripción detallada

3.18.2. Documentación del constructor y destructor

3.18.2.1. Radar::Radar (Frame * *frame*)

Instancia del radar

Hace referencia a Frame::activarFrame(), Frame::getH(), Frame::getVentana(), Frame::getW(), Frame::getX(), y Frame::getY().

3.18.3. Documentación de las funciones miembro

3.18.3.1. void Radar::recargar (bool *refresh*)

Recarga el radar, 'refresh' a true si se precisa refrescar

Hace referencia a Frame::activarFrame(), Frame::getH(), Frame::getVentana(), Frame::getW(), Frame::getX(), y Frame::getY().

Referenciado por Pantalla::entrada().

3.18.3.2. int Radar::getX ()

Devuelve la componente X del radar

3.18.3.3. int Radar::getY ()

Devuelve la componente Y del radar

3.18.3.4. int Radar::getR1 ()

Devuelve el radio de la circunferencia más pequeña

3.18.3.5. int Radar::getR2 ()

Devuelve el radio de la circunferencia mediana

3.18.3.6. int Radar::getR3 ()

Devuelve el radio de la circunferencia más grande

3.18.3.7. Frame * Radar::getFrame ()

Devuelve el frame que contiene el radar
Referenciado por dibujarFlecha().

3.18.3.8. vector< Punto > Radar::getObstaculos ()

Devuelve la lista de obstáculos

3.18.3.9. void Radar::addObstaculo (Punto *o*)

Añadir un nuevo punto 'o'

3.18.3.10. void Radar::dibujarFlecha (int *rot*)

Dibuja la flecha que indica la rotación de la silla
Hace referencia a getFrame().

3.18.3.11. SDL_Surface * Radar::getFlecha ()

Devuelve el Surface donde se pinta la flecha

3.18.3.12. SDL_Rect Radar::getDesp ()

Devuelve el desplazamiento

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- /home/diego/proyecto/touchscreen/src/radar.h
- /home/diego/proyecto/touchscreen/src/radar.cpp

3.19. Referencia de la Clase Selector

Gestiona la selección de mapas de un dispositivo USB.

`#include <selector.h>`

Métodos públicos

- Selector (SDL_Surface *pantalla)
- void buscarW (string ruta, string ext)
- void buscarR ()
- void cargar ()
- bool vacio ()
- bool handle (int x, int y)

3.19.1. Documentación del constructor y destructor

3.19.1.1. Selector::Selector (SDL_Surface * *pantalla*)

Constructor de la clase

3.19.2. Documentación de las funciones miembro

3.19.2.1. void Selector::buscarW (string *ruta*, string *ext*)

Obtiene un listado de ficheros del path "ruta" con extensió "ext"

Referenciado por Pantalla::entrada().

3.19.2.2. void Selector::buscarR ()

Cargar el mapa en memoria

3.19.2.3. void Selector::cargar ()

Muestra la lista de ficheros obtenida

Hace referencia a Boton::cargarBoton(), Frame::cargarFrame(), Frame::getW(), Frame::getX(), Frame::getY(), y Pantalla::mapaOff().

3.19.2.4. bool Selector::vacio ()

Comprueba si la lista de ficheros está vacía

3.19.2.5. bool Selector::handle (int *x*, int *y*)

Comprueba que plano de la lista ha sido pulsado

Hace referencia a Mapa::lectura(), y Pantalla::minimizar().

Referenciado por Pantalla::entrada().

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- /home/diego/proyecto/touchscreen/src/selector.h
- /home/diego/proyecto/touchscreen/src/selector.cpp

3.20. Referencia de la Clase Silla

Gestiona la posición y orientación de la silla.

`#include <silla.h>`

Métodos públicos

- Silla (Frame *frame, Mapa *plano)
- Punto getPos ()
- void setPos (Punto pos)
- int getRot ()
- void setRot (int r)
- void toogleStatus ()
- int getStatus ()
- void dibujar ()

3.20.1. Descripción detallada

3.20.2. Documentación del constructor y destructor

3.20.2.1. Silla::Silla (Frame * *frame*, Mapa * *plano*)

Crear una silla en el Frame 'frame' en el Mapa 'plano'

3.20.3. Documentación de las funciones miembro

3.20.3.1. Punto Silla::getPos ()

Devuelve la posicion de la silla

3.20.3.2. void Silla::setPos (Punto *pos*)

Actualiza la posicion actual al punto 'pos'

3.20.3.3. int Silla::getRot ()

Devuelve la rotacion 0-360

3.20.3.4. void Silla::setRot (int *r*)

Actualiza la rotacion actual a 'r'

3.20.3.5. void Silla::toogleStatus ()

Cambia el estado de la silla (0=apagado 1=encendido)

Referenciado por Pantalla::entrada().

3.20.3.6. int Silla::getStatus ()

Obtiene el estado de la silla (0=apagado 1=encendido)

Referenciado por Pantalla::entrada().

3.20.3.7. void Silla::dibujar ()

Dibuja la silla en la posicion correspondiente

Hace referencia a Punto::cpantalla(), Frame::getArea(), Mapa::getDH(), Mapa::getDV(), Mapa::getEscala(), Frame::getVentana(), Punto::getX(), Punto::getY(), Mapa::pintarMapa(), y Frame::refrescarFrame().

Referenciado por Pantalla::entrada(), y Pantalla::minimizar().

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- /home/diego/proyecto/touchscreen/src/silla.h
- /home/diego/proyecto/touchscreen/src/silla.cpp

3.21. Referencia de la Clase Tabla

tabla compuesta por elementos de clase Campo

`#include <tabla.h>`

Métodos públicos

- void add (string n, Campo *c)
- Campo get (string n)
- void update (string n, string v)
- void update (string n, float v)
- void handle (int x, int y)
- void recargar (Frame *frame)

3.21.1. Descripción detallada

3.21.2. Documentación de las funciones miembro

3.21.2.1. void Tabla::add (string *n*, Campo * *c*)

Añadir un campo a la tabla con la etiqueta unica 'n'

3.21.2.2. Campo Tabla::get (string *n*)

Devuelve el campo con la etiqueta 'n'

3.21.2.3. void Tabla::update (string *n*, string *v*)

Cambia el valor del campo con la etiqueta 'n' por el valor string 'v'

3.21.2.4. void Tabla::update (string *n*, float *v*)

Cambia el valor del campo con la etiqueta 'n' por el valor float 'v'

3.21.2.5. void Tabla::handle (int *x*, int *y*)

Gestiona el input de la pantalla tactil

Referenciado por Pantalla::entrada().

3.21.2.6. void Tabla::recargar (Frame * *frame*)

Recarga la tabla en el Frame 'frame'

Hace referencia a Frame::getEstado(), Frame::getH(), Frame::getX(), y Frame::getY().

Referenciado por Pantalla::entrada(), y Pantalla::minimizar().

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- /home/diego/proyecto/touchscreen/src/tabla.h
- /home/diego/proyecto/touchscreen/src/tabla.cpp

Capítulo 4

Documentación de archivos

4.1. Referencia del Archivo /home/diego/proyecto/touchscreen/src/boton.cpp

```
#include "boton.h"
#include "constantes.h"
#include <string>
#include <SDL/SDL_gfxPrimitives.h>
#include <iostream>
#include <SDL/SDL_rotozoom.h>
```

4.1.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.2. Referencia del Archivo /home/diego/proyecto/touchscreen/src/boton.h

```
#include <SDL/SDL.h>
#include <iostream>
```

Clases

- class Boton

  Gestiona la entidad botón.

Enumeraciones

- enum EstadoBoton { **activo** = 1, **inactivo** = 0, **z** = 3 }

4.2.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.2.2. Documentación de las enumeraciones

4.2.2.1. enum EstadoBoton

El botó puede estar el estado: activo o inactivo

4.3. Referencia del Archivo /home/diego/proyecto/touchscreen/src/campo.cpp

```
#include "campo.h"
#include "constantes.h"
#include <sstream>
#include <iostream>
#include <SDL/SDL_gfxPrimitives.h>
```

4.3.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.4. Referencia del Archivo /home/diego/proyecto/touchscreen/src/campo.h

```
#include <SDL/SDL.h>
#include <string>
#include "boton.h"
#include <iostream>
```

Clases

- class Campo

  Los elementos campo almacenan información estática o numérica modificable.

4.4.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.5. Referencia del Archivo /home/diego/proyecto/touchscreen/src/capa.cpp

```
#include "capa.h"
```

4.5.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.6. Referencia del Archivo /home/diego/proyecto/touchscreen/src/capa.h

```
#include <iostream>
#include <vector>
#include "linea.h"
#include "polilinea.h"
#include "punto.h"
```

Clases

- class Capa

  La clase Mapa está dividida en capas.

4.6.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.7. Referencia del Archivo /home/diego/proyecto/touchscreen/src/clientecapa_alta.cpp

```
#include "clientecapa_alta.h"
#include <SDL/SDL_thread.h>
#include <sstream>
#include <iostream>
#include "tabla.h"
#include "constantes.h"
```

Funciones

- int **conectarAlta** (void *data)

Variables

- SDL_mutex * **mutexCapaAlta**
- SDL_cond * **caminoNuevoCond**

4.7.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.8. Referencia del Archivo /home/diego/proyecto/touchscreen/src/clientealta.h

```
#include <SolarSockets/SolarSockets++.h>
#include <fstream>
#include <sstream>
#include <iostream>
#include <vector>
#include "punto.h"
```

Clases

- class ClienteCapaAlta

  Gestiona la comunicación con el sistema IA. Información aperiódica.

4.8.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.9. Referencia del Archivo /home/diego/proyecto/touchscreen/src/clientecapa_baja.cpp

```
#include "clientecapa_baja.h"
#include <SDL/SDL.h>
#include "tabla.h"
#include "silla.h"
#include "constantes.h"
```

Funciones

- int **conectarBaja** (void *data)

Variables

- SDL_cond * **sensorNuevoCond**
- SDL_mutex * **mutexCapaBaja**

4.9.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.10. Referencia del Archivo /home/diego/proyecto/touchscreen/src/clien baja.h

```
#include <SolarSockets/SolarSockets++.h>
#include <fstream>
#include <vector>
#include <SDL/SDL_thread.h>
#include <sstream>
#include <iostream>
#include "punto.h"
```

Clases

- class ClienteCapaBaja

  Gestiona la comunicacin con el sistema IA. Información periódica.

4.10.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.11. Referencia del Archivo /home/diego/proyecto/touchscreen/src/constantes.h

Definiciones

- #define **COLOR_BG** 0x000000
- #define **COLOR_BORDER_BOTON** 0xFFFFFFFF
- #define **COLOR_FUENTE** 0x0066FFFF
- #define **COLOR_BORDER_FRAME** 0x0066FFFF
- #define **SCREEN_W** 640
- #define **SCREEN_H** 480
- #define **BORDER** 2
- #define **MARGENH** 20
- #define **MARGENV** 30
- #define **CERRADO** 0
- #define **MAXIMO** 1
- #define **MINIMO** 2
- #define **SIZE_C** 8
- #define **FACTOR_ZOOM** 50
- #define **FACTOR_DESP** 10
- #define **ZOOM_MAX** 1500
- #define **ZOOM_MIN** 25
- #define **T_BOTON** 20
- #define **Z_ARRIBA** 0x1
- #define **Z_IZQUIERDA** 0x2
- #define **Z_ABAJO** 0x4
- #define **Z_DERECHA** 0x8
- #define **Z_CENTRO** 0x10
- #define **Z_TOTAL** 0x1F
- #define **TRAMAPLANO** 0
- #define **TRAMAOBJETIVO** 1
- #define **CLIENTESENS** 2
- #define **SIN_RESPUESTA** 0
- #define **RESPUESTA_SI** 1
- #define **RESPUESTA_NO** 2
- #define **COLORCAMINO** 0xff0000ff
- #define **CABECERA_MAPA** "[MAPA] "
- #define **CABECERA_INIRUTA** "[RUTA] "
- #define **CABECERA_FINRUTA** "[FIRU] "
- #define **CABECERA_ERROR** "[ERRO] "
- #define **CABECERA_STATUS** "[STAT] "
- #define **CABECERA_OBJETIVO** "[OBJE] "
- #define **CABECERA_SENS** "[SENS] "
- #define **CABECERA_POS** "[POSI] "
- #define **CABECERA_GRID** "[GRID] "
- #define **CABECERA_DOBST** "[DOBS] "
- #define **CABECERA_FLAGS** "[FLAG] "

4.11.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.12. Referencia del Archivo /home/diego/proyecto/touchscreen/src/dibujar.cpp

```
#include "dibujar.h"
#include "constantes.h"
```

4.12.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.13. Referencia del Archivo /home/diego/proyecto/touchscreen/src/dibujar.h

```
#include <SDL/SDL.h>
#include <SDL/SDL_gfxPrimitives.h>
#include "linea.h"
#include "frame.h"
```

Clases

- class Dibujar

  Representación gráfica del Mapa.

4.13.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.14. Referencia del Archivo /home/diego/proyecto/touchscreen/src/dxfparser.cpp

```
#include "dxfparser.h"
#include <iostream>
#include <stdlib.h>
#include "mapa.h"
#include "linea.h"
#include "capa.h"
#include "polilinea.h"
#include "punto.h"
```

Variables

- Mapa **plano**

4.14.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.15. Referencia del Archivo /home/diego/proyecto/touchscreen/src/dxfparser.h

```
#include <dxflib/dl_creationadapter.h>
```

Clases

- class DxfParser

  Parseador para el fichero DXF.

4.15.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.16. Referencia del Archivo /home/diego/proyecto/touchscreen/src/etiqueta.cpp

```
#include "etiqueta.h"
#include "constantes.h"
#include <string>
#include <SDL/SDL_gfxPrimitives.h>
```

4.16.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.17. Referencia del Archivo /home/diego/proyecto/touchscreen/src/etiqueta.h

```
#include <SDL/SDL.h>
#include <iostream>
```

Clases

- class Etiqueta

  Gestiona etiquetas de texto.

4.17.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.18. Referencia del Archivo /home/diego/proyecto/touchscreen/src/frame.cpp

```
#include "frame.h"
#include "constantes.h"
#include "dibujar.h"
#include "etiqueta.h"
#include <SDL/SDL_gfxPrimitives.h>
```

4.18.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.19. Referencia del Archivo /home/diego/proyecto/touchscreen/src/frame.h

```
#include <SDL/SDL.h>
#include "boton.h"
#include <iostream>
```

Clases

- class Frame

  Gestiona el sistema de ventanas (Frames).

4.19.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.20. Referencia del Archivo /home/diego/proyecto/touchscreen/src/gestorcamino.cpp

```
#include <iostream>
#include "gestorcamino.h"
#include <vector>
#include "punto.h"
#include "clientecapa_alta.h"
#include "constantes.h"
#include "frame.h"
#include "polilinea.h"
#include "mapa.h"
#include "pantalla.h"
#include "etiqueta.h"
#include <algorithm>
#include "silla.h"
```

Funciones

- int **threadCamino** (void *p)

Variables

- ClienteCapaAlta **clienteCapaAlta**
- SDL_mutex * **mutexCapaAlta**
- SDL_cond * **caminoNuevoCond**
- Mapa **plano**
- Pantalla * **pantalla**
- Silla * **silla**

4.20.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.21. Referencia del Archivo /home/diego/proyecto/touchscreen/src/gestorcamino.h

```
#include <SDL/SDL_thread.h>
#include <SDL/SDL.h>
#include <stdlib.h>
#include "frame.h"
```

Clases

- class GestorCamino

  Hilo que se encarga de la lectura del camino cuando se recibe.

4.21.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.22. Referencia del Archivo /home/diego/proyecto/touchscreen/src/gestorestado.cpp

```
#include "gestorestado.h"
```

Funciones

- int **threadEstado** (void *p)

Variables

- SDL_cond * **sensorNuevoCond**
- SDL_mutex * **mutexCapaBaja**
- ClienteCapaBaja **clienteCapaBaja**

4.22.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.23. Referencia del Archivo /home/diego/proyecto/touchscreen/src/gestorestado.h

```
#include <SDL/SDL_thread.h>
#include <SDL/SDL.h>
#include <stdlib.h>
#include <iostream>
#include <vector>
#include "clientecapa_baja.h"
```

Clases

- class GestorEstado

  Hilo que realiza la lectura de los valores de los sensores cuando se reciben.

4.23.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.24. Referencia del Archivo /home/diego/proyecto/touchscreen/src/gui.cpp

```
#include <iostream>
#include <stdlib.h>
#include <SDL/SDL.h>
#include <SDL/SDL_thread.h>
#include <SDL/SDL_gfxPrimitives.h>
#include <dxflib/dl_dxf.h>
#include "pantalla.h"
#include "boton.h"
#include "etiqueta.h"
#include "frame.h"
#include "constantes.h"
#include "dxfparser.h"
#include "mapa.h"
#include "campo.h"
#include "objetivo.h"
#include "clientecapa_alta.h"
#include "clientecapa_baja.h"
#include "gestorestado.h"
#include "gestorcamino.h"
#include "radar.h"
#include "tabla.h"
#include "silla.h"
#include "selector.h"
#include <pthread.h>
```

Funciones

- int **main** (int argc, char *argv[ ])

Variables

- SDL_Surface * **surfacePrincipal**
- Pantalla * **pantalla**

- Boton * **botonMasZoom**
- Boton * **botonMenosZoom**
- Boton * **botonAjustarZoom**
- Boton * **botonArriba**
- Boton * **botonAbajo**
- Boton * **botonDerecha**
- Boton * **botonIzquierda**
- Boton * **botonCentrar**
- Boton * **botonSelector**
- Boton * **botonOnoff**
- Frame * **framemapa**
- Frame * **frameradar**
- Frame * **framestado**
- Frame * **frameselector**
- Etiqueta * **e_zoom**
- Mapa **plano**
- Objetivo **objetivo**
- Radar * **radar**
- Silla * **silla**
- Selector * **selector**
- SDL_mutex * **mutexCapaAlta**
- SDL_cond * **caminoNuevoCond**
- SDL_mutex * **mutexCapaBaja**
- SDL_cond * **sensorNuevoCond**
- SDL_mutex * **mutexSincRadar**
- SDL_cond * **condSincRadar**
- bool **pauseRadar**
- SDL_mutex * **semVideo**
- SDL_mutex * **mutexObstaculo**
- SDL_mutex * **mutexRot**
- Campo * **cconex**
- Campo * **cgrid**
- Campo * **cdobstaculo**
- Campo * **cvelocidad**
- Tabla **tcampos**
- ClienteCapaAlta **clienteCapaAlta**
- ClienteCapaBaja **clienteCapaBaja**

4.24.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.25. Referencia del Archivo /home/diego/proyecto/touchscreen/src/linea.cpp

```
#include "linea.h"
```

4.25.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.26. Referencia del Archivo /home/diego/proyecto/touchscreen/src/linea.h

```
#include <dxflib/dl_dxf.h>
#include "punto.h"
```

Clases

- class Linea

  linea del mapa asignada a una capa

4.26.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.27. Referencia del Archivo /home/diego/proyecto/touchscreen/src/mapa.cpp

```
#include "mapa.h"
#include "constantes.h"
#include <dxflib/dl_dxf.h>
#include "dxfparser.h"
#include <math.h>
```

4.27.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.28. Referencia del Archivo /home/diego/proyecto/touchscreen/src/mapa.h

```
#include <vector>
#include "capa.h"
#include "frame.h"
#include "dibujar.h"
```

Clases

- class Mapa

  Organiza las capas del fichero dxf.

4.28.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.29. Referencia del Archivo /home/diego/proyecto/touchscreen/src/objetivo.cpp

```
#include "objetivo.h"
#include "linea.h"
#include "polilinea.h"
#include "constantes.h"
#include <SDL/SDL_gfxPrimitives.h>
#include <SDL/SDL.h>
#include <iostream>
#include <sstream>
```

4.29.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.30. Referencia del Archivo /home/diego/proyecto/touchscreen/src/objetivo.h

```
#include "polilinea.h"
#include "frame.h"
#include "mapa.h"
#include "boton.h"
```

Clases

- class Objetivo

  Clase que gestiona el objetivo establecido.

4.30.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.31. Referencia del Archivo /home/diego/proyecto/touchscreen/src/pantalla.cpp

```
#include "pantalla.h"
#include "constantes.h"
#include "boton.h"
#include "mapa.h"
#include "etiqueta.h"
#include "tabla.h"
#include "objetivo.h"
#include "radar.h"
#include <SDL/SDL_thread.h>
#include <sstream>
#include <iostream>
#include "clientecapa_alta.h"
#include "silla.h"
#include "selector.h"
```

Variables

- Boton * **botonMasZoom**
- Boton * **botonMenosZoom**
- Boton * **botonAjustarZoom**
- Boton * **botonDerecha**
- Boton * **botonIzquierda**
- Boton * **botonArriba**
- Boton * **botonAbajo**
- Boton * **botonCentrar**
- Boton * **botonSelector**
- Boton * **botonOnoff**
- Mapa **plano**
- Frame * **framemapa**
- Frame * **frameradar**
- Frame * **framestado**
- Frame * **frameselector**
- Etiqueta * **e_zoom**
- Objetivo **objetivo**
- ClienteCapaAlta **clienteCapaAlta**
- Radar * **radar**
- SDL_mutex * **mutexSincRadar**

- SDL_cond * **condSincRadar**
- bool **pauseRadar**
- Tabla **tcampos**
- Silla * **silla**
- Selector * **selector**

4.31.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.32. Referencia del Archivo /home/diego/proyecto/touchscreen/src/pantalla.h

```
#include <iostream>
#include <stdlib.h>
#include <SDL/SDL.h>
#include "frame.h"
#include "punto.h"
```

Clases

- class Pantalla

  Gestiona todos los aspectos relacionados con el monitor (como la resolución).

4.32.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.33. Referencia del Archivo /home/diego/proyecto/touchscreen/src/polilinea.cpp

```
#include "polilinea.h"
```

4.33.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.34. Referencia del Archivo /home/diego/proyecto/touchscreen/src/polilinea.h

```
#include <iostream>
#include <vector>
#include "punto.h"
#include "linea.h"
```

Clases

- class Polilinea

  Elemento formado por un conjunto de Puntos.

4.34.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.35. Referencia del Archivo /home/diego/proyecto/touchscreen/src/punto.cpp

```
#include "punto.h"
```

4.35.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.36. Referencia del Archivo /home/diego/proyecto/touchscreen/src/punto.h

```
#include <dxflib/dl_dxf.h>
#include "frame.h"
```

Clases

- class Punto

  Punto de coordenadas (x,y).

4.36.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.37. Referencia del Archivo /home/diego/proyecto/touchscreen/src/radar.cpp

```
#include "radar.h"
#include <SDL/SDL_gfxPrimitives.h>
#include "constantes.h"
#include <SDL/SDL_rotozoom.h>
#include "silla.h"
```

Funciones

- int **escan** (void *r)

Variables

- SDL_mutex * **bloqpantalla**

4.37.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.38. Referencia del Archivo /home/diego/proyecto/touchscreen/src/radar.h

```
#include "punto.h"
#include "frame.h"
#include <vector>
#include <SDL/SDL_thread.h>
```

Clases

- class Radar

  El radar indica los obstáculos próximos y la orientación de la silla.

4.38.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.39. Referencia del Archivo /home/diego/proyecto/touchscreen/src/selector.cpp

```
#include "selector.h"
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/wait.h>
#include <fcntl.h>
#include <iostream>
#include "frame.h"
#include "constantes.h"
#include "mapa.h"
#include "pantalla.h"
#include "etiqueta.h"
```

Funciones

- void **reaper** (int iSignal)

4.39.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.40. Referencia del Archivo /home/diego/proyecto/touchscreen/src/selector.h

```
#include <string>
#include <vector>
#include <SDL/SDL.h>
#include "boton.h"
```

Clases

- class Selector

  Gestiona la selección de mapas de un dispositivo USB.

4.40.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.41. Referencia del Archivo /home/diego/proyecto/touchscreen/src/silla.cpp

```
#include "silla.h"
```

4.41.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.42. Referencia del Archivo /home/diego/proyecto/touchscreen/src/silla.h

```
#include "punto.h"
#include "frame.h"
#include "mapa.h"
```

Clases

- class Silla

  Gestiona la posición y orientación de la silla.

4.42.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.43. Referencia del Archivo /home/diego/proyecto/touchscreen/src/tabla.cpp

```
#include "tabla.h"
#include "constantes.h"
```

4.43.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com

4.44. Referencia del Archivo /home/diego/proyecto/touchscreen/src/tabla.h

```
#include <map>
#include <utility>
#include <string>
#include "campo.h"
#include "frame.h"
#include <iostream>
#include "clientecapa_alta.h"
```

Clases

- class Tabla

  tabla compuesta por elementos de clase Campo

4.44.1. Descripción detallada

Autor:

Diego García , kobydiego@gmail.com