

My Project

Generated by Doxygen 1.8.17

Chapter 1

Class Index

1.1 Class List

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

acumulador	T.D.A. Acumulador	??
cola	T.D.A. Cola	??
ficha	T.D.A. Ficha	??
juego	??

Chapter 2

File Index

2.1 File List

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

acumulador.h	??
cola.h	??
ficha.h	??
juego.h	??

Chapter 3

Class Documentation

3.1 acumulador Class Reference

T.D.A. Acumulador.

```
#include <acumulador.h>
```

Public Member Functions

- [acumulador](#) ()
constructor base
- [acumulador](#) (const [acumulador](#) &orig)
constructor copia
- [acumulador](#) (int h, int w)
constructor con dimensiones
- [~acumulador](#) ()
destructor de la clase
- void [dejarCaer](#) ([ficha](#) f)
coloca la ficha para caer arriba del tablero
- bool [finDePartida](#) () const
indica si la partida ha acabado
- void [rotarFicha](#) (bool horario)
rota la ficha actual
- void [moverFicha](#) (bool derecha)
mueve la ficha actual
- void [depositarFicha](#) ()
deposita la ficha actual
- bool [puedeCaer](#) ()
determina si la ficha puede seguir cayendo o esta depositada
- void [ticDeReloj](#) ()
baja la ficha una casilla
- bool [filaLlena](#) () const
comprueba si hay filas llenas
- int [limpiarFilas](#) ()
borra las filas llenas

Friends

- `std::ostream & operator<< (std::ostream &os, const acumulador &ac)`
imprime el tablero

3.1.1 Detailed Description

T.D.A. Acumulador.

Una instancia del tipo acumulador representa el tablero donde vamos a jugar. Se encarga de almacenar las piezas depositadas durante la partida. También almacena la ficha que estamos jugando en un determinado instante de tiempo.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 acumulador() [1/3]

```
acumulador::acumulador ( )
```

constructor base

Note

inicializa el tablero a `ficha::tipo::vacio`

See also

[ficha::tipo](#)

3.1.2.2 acumulador() [2/3]

```
acumulador::acumulador (
    const acumulador & orig )
```

constructor copia

Parameters

<i>orig</i>	instancia de la clase acumulador
-------------	----------------------------------

3.1.2.3 acumulador() [3/3]

```
acumulador::acumulador (
    int h,
    int w )
```

constructor con dimensiones

Parameters

<i>h</i>	altura del tablero
<i>w</i>	anchura del tablero

Note

inicializa el tablero con esas dimensiones a `ficha::tipo::vacio`

See also

[ficha::tipo](#)

3.1.3 Member Function Documentation

3.1.3.1 dejarCaer()

```
void acumulador::dejarCaer (
    ficha f )
```

coloca la ficha para caer arriba del tablero

Parameters

<i>f</i>	la ficha que se deja caer
----------	---------------------------

Note

la coloca en el centro de la parte superior del tablero

3.1.3.2 depositarFicha()

```
void acumulador::depositarFicha ( )
```

deposita la ficha actual

Warning

usar para bajar la ficha directamente

3.1.3.3 filaLlena()

```
bool acumulador::filaLlena ( ) const
```

comprueba si hay filas llenas

Return values

<i>true</i>	si hay filas llenas
-------------	---------------------

3.1.3.4 finDePartida()

```
bool acumulador::finDePartida ( ) const
```

indica si la partida ha acabado

Return values

<i>true</i>	si la partida ha acabado(no se puede dejar caer otra ficha)
-------------	---

3.1.3.5 limpiarFilas()

```
int acumulador::limpiarFilas ( )
```

borra las filas llenas

Returns

la cantidad de filas eliminadas

Note

usar con [acumulador::filaLlena\(\)](#)

See also

[acumulador::filaLlena\(\)](#)

3.1.3.6 moverFicha()

```
void acumulador::moverFicha (
    bool derecha )
```

mueve la ficha actual

Parameters

<i>derecha</i>	mueve la ficha a la derecha si es true. En caso contrario hacia la izquierda
----------------	--

Note

si la ficha no puede ser movida se mantendrá como está

3.1.3.7 puedeCaer()

```
bool acumulador::puedeCaer ( )
```

determina si la ficha puede seguir cayendo o esta depositada

Return values

<i>true</i>	si la ficha puede seguir cayendo
-------------	----------------------------------

Note

en caso de que la ficha este depositada se cambiaran los tipo::activo del tablero por tipo::forma

3.1.3.8 rotarFicha()

```
void acumulador::rotarFicha (
    bool horario )
```

rota la ficha actual

Parameters

<i>horario</i>	decide si rotar la ficha en sentido horario o antihorario
----------------	---

Note

si la ficha no puede ser rotada se mantendrá como está

3.1.3.9 ticDeReloj()

```
void acumulador::ticDeReloj ( )
```

baja la ficha una casilla

Warning

llamar periodicamente con un reloj, aumentar el reloj para mayor dificultad

3.1.4 Friends And Related Function Documentation

3.1.4.1 operator<<

```
std::ostream& operator<< (
    std::ostream & os,
    const acumulador & ac ) [friend]
```

imprime el tablero

Parameters

<i>os</i>	instancia clase ostream
-----------	-------------------------

See also

[std::ostream](#)

Parameters

<i>ac</i>	instancia clase acumulador
<i>os</i>	instancia clase ostream

See also

[std::ostream](#)

Parameters

<i>ac</i>	instancia clase acumulador
-----------	----------------------------

Warning

nota para el profesor, no usamos clase imagen ya que como hemos implementado el acumulador solo hace falta "colorear", es decir, asociar una imagen nxn a cada [ficha::tipo](#)

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

- [acumulador.h](#)

3.2 cola Class Reference

T.D.A. Cola.

```
#include <cola.h>
```

Public Member Functions

- [cola](#) ()
constructor de cola
- [cola](#) (const [cola](#) &orig)
constructor copia
- [~cola](#) ()
destructor de la clase
- `std::vector< ficha > getFichas ()` const
devuelve un vector con las componentes de la lista
- [ficha sacarFicha](#) ()
devuelve el primer elemento de la lista, lo devuelve, lo elimina y añade uno nuevo al final

Friends

- `std::ostream & operator<< (std::ostream &os, const cola &cl)`
imprime el cola

3.2.1 Detailed Description

T.D.A. Cola.

Una instancia del tipo de dato abstracto cola es un objeto que contiene las 4 siguientes fichas a jugar

3.2.2 Constructor & Destructor Documentation

3.2.2.1 [cola\(\)](#) [1/2]

```
cola::cola ( )
```

constructor de cola

Note

genera 4 fichas aleatorias y las mete en la cola

3.2.2.2 [cola\(\)](#) [2/2]

```
cola::cola (
    const cola & orig )
```

constructor copia

Parameters

<i>orig</i>	instancia clase cola
-------------	----------------------

3.2.3 Member Function Documentation

3.2.3.1 getFichas()

```
std::vector<ficha> cola::getFichas ( ) const
```

devuelve un vector con las componentes de la lista

Returns

std::vector con todas las componentes de la lista

3.2.3.2 sacarFicha()

```
ficha cola::sacarFicha ( )
```

devuelve el primer elemento de la lista, lo devuelve, lo elimina y añade uno nuevo al final

Returns

la primera ficha de la lista

3.2.4 Friends And Related Function Documentation

3.2.4.1 operator<<

```
std::ostream& operator<< (
    std::ostream & os,
    const cola & cl ) [friend]
```

imprime el cola

Parameters

<i>os</i>	instancia clase ostream
-----------	-------------------------

See also

`std::ostream`

Parameters

<i>cl</i>	instancia clase cola
-----------	----------------------

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

- [cola.h](#)

3.3 ficha Class Reference

T.D.A. Ficha.

```
#include <ficha.h>
```

Public Types

- enum [tipo](#) {
vacio, **recta**, **cuadrado**, **t**,
IDerecha, **Ilzquierda**, **zDerecha**, **zlzquierda**,
activo }

enum indicando el tipo de ficha

Public Member Functions

- [ficha](#) ()
constructor base
- [ficha](#) (const [ficha](#) &orig)
constructor copia
- [ficha](#) ([ficha::tipo](#) forma)
constructor inializando la forma
- [~ficha](#) ()
destructor de la clase
- void [setForma](#) ([ficha::tipo](#) forma)
establece la forma de la ficha
- [tipo](#) [getForma](#) () const
devuelve la forma establecida
- `std::vector< std::vector< ficha::tipo > > getMatriz ()` const
devuelve la matriz asociada a la forma de la ficha

3.3.1 Detailed Description

T.D.A. Ficha.

Una instancia del tipo de dato abstracto Ficha es un objeto que representa una ficha del tetris. Está representado por una forma seleccionada de un enum.

3.3.2 Member Enumeration Documentation

3.3.2.1 tipo

```
enum ficha::tipo
```

enum indicando el tipo de ficha

Enumerator

activo	
	Warning
	Solo usado por acumulador

3.3.3 Constructor & Destructor Documentation

3.3.3.1 `ficha()` [1/2]

```
ficha::ficha (
    const ficha & orig )
```

constructor copia

Parameters

<i>orig</i>	instancia clase ficha
-------------	-----------------------

3.3.3.2 `ficha()` [2/2]

```
ficha::ficha (
    ficha::tipo forma )
```

constructor inializando la forma

Parameters

<i>forma</i>	tipo de la ficha
--------------	------------------

Note

se recomienda usar

3.3.4 Member Function Documentation

3.3.4.1 getForma()

```
tipo ficha::getForma ( ) const
```

devuelve la forma establecida

Returns

la forma de la ficha

3.3.4.2 getMatriz()

```
std::vector<std::vector<ficha::tipo> > ficha::getMatriz ( ) const
```

devuelve la matriz asociada a la forma de la ficha

Returns

la matriz asociada a la forma de la ficha

Note

en la matriz solo se marcan las casillas ocupadas y las no ocupadas tendran el tipo tipo::vacio

See also

[ficha::tipo](#)

3.3.4.3 setForma()

```
void ficha::setForma (
    ficha::tipo forma )
```

establece la forma de la ficha

Parameters

<i>forma</i>	la forma a establecer
--------------	-----------------------

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

- [ficha.h](#)

3.4 juego Class Reference

Public Member Functions

- void [print](#) () const
metodo alternativo de impresion ya sea porque se desea ventana aparte y no querer usar la salida estandar
- [juego](#) ()
constructor base
- [juego](#) (const [juego](#) &orig)
constructor copia
- [~juego](#) ()
destructor de la clase
- void [jugar](#) ()
comienza y desarrolla una partida de tetris

Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [juego](#) &jg)
imprime el juego

3.4.1 Member Function Documentation

3.4.1.1 jugar()

```
void juego::jugar ( )
```

comienza y desarrolla una partida de tetris

Warning

aqui se incluye toda la lógica del juego
finaliza si se fuerza la finalización o si jugando es false(has perdido)

3.4.1.2 print()

```
void juego::print ( ) const
```

metodo alternativo de impresion ya sea porque se desea ventana aparte y no querer usar la salida estandar

Warning

implementación libre y equivale a la clase Imagen

3.4.2 Friends And Related Function Documentation

3.4.2.1 operator<<

```
std::ostream& operator<< (
    std::ostream & os,
    const juego & jg ) [friend]
```

imprime el juego

Parameters

<i>os</i>	instancia clase ostream
-----------	-------------------------

See also

`std::ostream`

Parameters

<i>jg</i>	instancia clase juego
-----------	-----------------------

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

- [juego.h](#)

Chapter 4

File Documentation

4.1 acumulador.h File Reference

```
#include <ostream>
#include "ficha.h"
Include dependency graph for acumulador.h:
```

4.2 cola.h File Reference

```
#include <ostream>
#include "ficha.h"
Include dependency graph for cola.h: This graph shows which files directly or indirectly include this file:
```

Classes

- class [cola](#)
T.D.A. Cola.

Functions

- `std::ostream & operator<< (std::ostream &os, const cola &c)`
imprime el cola

4.2.1 Detailed Description

Author

Mario García Márquez y Daniel Alconchel Vázquez

Date

13 de Octubre de 2020

4.2.2 Function Documentation

4.2.2.1 operator<<()

```
std::ostream& operator<< (
    std::ostream & os,
    const cola & cl )
```

imprime el cola

Parameters

<i>os</i>	instancia clase ostream
-----------	-------------------------

See also

std::ostream

Parameters

<i>cl</i>	instancia clase cola
-----------	----------------------

4.3 ficha.h File Reference

```
#include <vector>
```

Include dependency graph for ficha.h: This graph shows which files directly or indirectly include this file:

Classes

- class [ficha](#)
T.D.A. Ficha.

4.3.1 Detailed Description

Author

Mario García Márquez y Daniel Alconchel Vázquez

Date

13 de Octubre de 2020

4.4 juego.h File Reference

```
#include "acumulador.h"
#include "cola.h"
Include dependency graph for juego.h:
```

Classes

- class [juego](#)

Functions

- `std::ostream & operator<< (std::ostream &os, const juego &jg)`
imprime el juego

4.4.1 Detailed Description

Author

Mario García Márquez y Daniel Alconchel Vázquez

Date

13 de Octubre de 2020

4.4.2 Function Documentation

4.4.2.1 `operator<<()`

```
std::ostream& operator<< (
    std::ostream & os,
    const juego & jg )
```

imprime el juego

Parameters

<code>os</code>	instancia clase ostream
-----------------	-------------------------

See also

`std::ostream`

Parameters

<i>jg</i>	instancia clase juego
-----------	-----------------------