Mundo

*Variables*

x\_ojo, y\_ojo, z\_ojo

*Métodos*

Tecla (unsigned char key): vacío

Inicializa: x\_ojo, y\_ojo, z\_ojo = {0,10,20}

rotarOjo: rota pto vista

mueve: vacío

dibuja: dibuja el mundo con pto de vista x\_ojo, y\_ojo, z\_ojo y general un plano

Pieza

*Variables*

int fila, columna (matriz tablero)

bool next, ocupada (movimientos)

bool color (1 blanca)

int tipo

*Métodos*

Pieza(), virtual ~Pieza() (constructor y destructor).

Pieza(): ocupado y next falso, tipo 0.

virtual ~Pieza(): vacío

Dibuja(): Vacío

bool getColor

setPosition(float ix, float iy): vacío

setColor(unsigned char r, unsigned char v, unsigned char a): vacío

toggleOcupado: cambia si ocupado o no ocupado y vicerversa

setNext:vacío

setColor: Vacío

getPosicionX

getPosicionY

getOcupado

bool getNext

Vector2D

*Variables*

float x, y (coordenadas pieza)

*Métodos*

Vector2D, virtual ~Vector2D() (Constructor y destructor)

Vector2D(float ix, float iy): Sirve para poder declarar en tablero una matriz 8x8 con sus coordenadas asignadas

setPosicion(float ix,float iy)

float getX|float getY

Tablero

*Variables*

Vector2D casilla [8][8]: declara una matriz de 8x8, con coordenadas x,y asignadas de tipo casilla

Pieza \*id[8][8]: puntero tipo Pieza a direcciones de memoria. Si id=0 la casilla está vacía

bool turno

*Métodos*

Tablero() (constructor): matriz 8x8 de forma: casilla 1 es 10.0fx10.0f, casilla 2 es 20.0fx20.0f. todas las id asociadas son nulas, turno = 0

bool selPieza(int foreign, int corigen): si el id es 1

bool mover(int fdestino, int cdestino)

Peon **(hereda Pieza)**

*Variables*

bool mov\_ini. True si es el primer movimiento de la partida

*Métodos*

Peon(bool color, int fila, int columna): true si mov permido, la n de x e y es porque es el next move

bool mover(int nx, int ny) true si se puede comer

Rey **(hereda Pieza)**

*Métodos*

Rey(bool color, int fila, int columna)

void Dibuja();

bool mRey (float nx, float ny);