

rotaciones? [Versión Grilla]

La GUI capta el Event de que se presionó una tecla asociada a rotación (Tentativamente z, x). Para el ejemplo, suponemos que se rota hacia la derecha, pero para la izquierda es análogo.

Se comunica con Juego notificandole esto con el metodo pedirRotacionDerecha(). Juego se lo comunica a Grilla esto con el método rotarDerecha().

Grilla le pide a Tetrimino las Posiciones de los Bloques en los que este quedará después de la rotación, y estudia si están libres, accediendo a los Bloques a partir de las coordenadas. Si están libres, Grilla efectúa la rotación *setteando* Tetrimino los Bloques correspondientes a la nueva "forma" (Ver nota "¿Como pensamos los movimientos?") y cambiando el ángulo actual.

Grilla le notifica a Juego que Bloques fueron cambiados y con que imágenes se deberían representar después del cambio. Juego le notifica a la GUI que se deben actualizar las imágenes en las *JLabels* correspondientes a esos Bloques.

¿Como pensamos los movimientos? [Version Grilla]

La GUI capta el Event de que se presionó una tecla correspondiente a un movimiento o el Reloj marca un tick.

Se comunican con juego notificándole que hay que hacer un movimiento. Juego le notifica a Grilla que hay que hacer un movimiento.

Grilla le pide Tetrimino las Posiciones de los Bloques en los que este quedará después en los que este quedará después del movimiento, y estudia si estan libres. Si están libres, Grilla efectúa el movimiento *setteando* en Tetrimino los Bloques correspondientes a la nueva posición (Esto es, se cambian las referencias almacenadas en bloqueA, bloqueB, etc.), cambiando los estados y las imagenes donde sea necesario, etc.

Grilla le notifica a Juego que bloques fueron cambiados y con que imagenes se deberian representar despues del cambio. Juego le notifica a la GUI que se deben actualizar las imágenes en las *JLabels* correspondientes a esos bloques.

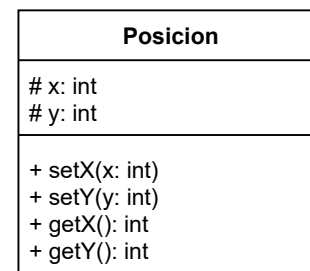
¿Como manejamos las colisiones? [Grilla]

Grilla ejecutó bloquesLibres() y obtuvo false en una llamada a moverAbajo(), por lo que el Tetrimino debe quedarse fijo.

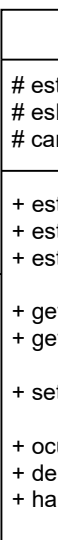
Grilla hace estatico el Tetrimino, llamando a hacerEstatico().

Grilla comprueba de que no se haya perdido. Si se perdio, envia le notifica a juego con el metodo finalizarPartida() y ese es el fin del manejo de la colision.

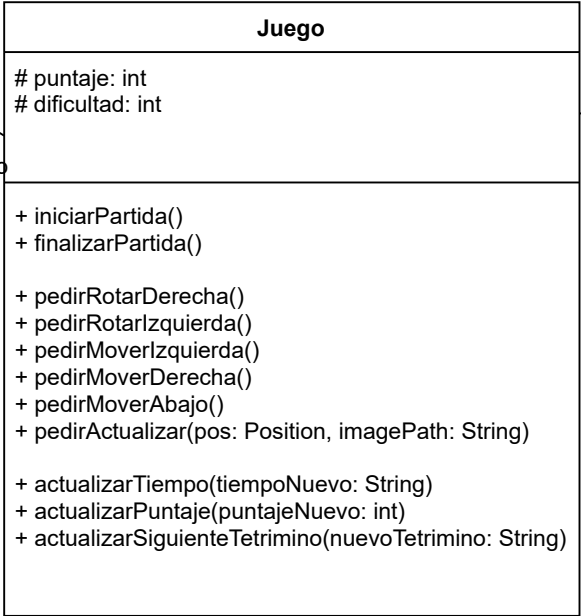
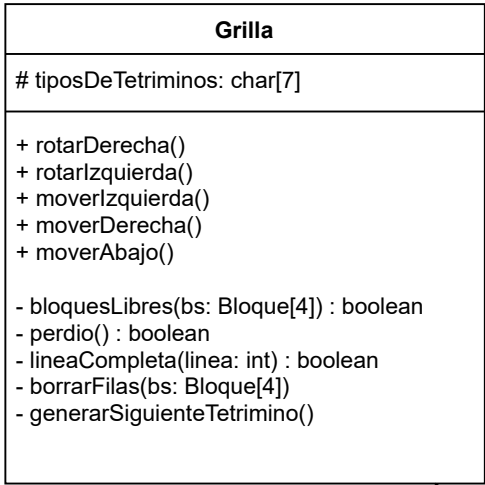
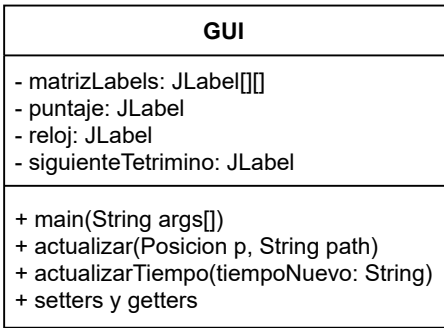
Si no se perdió, grilla checkea si se completaron filas, chequeando las 4 filas correspondientes a los



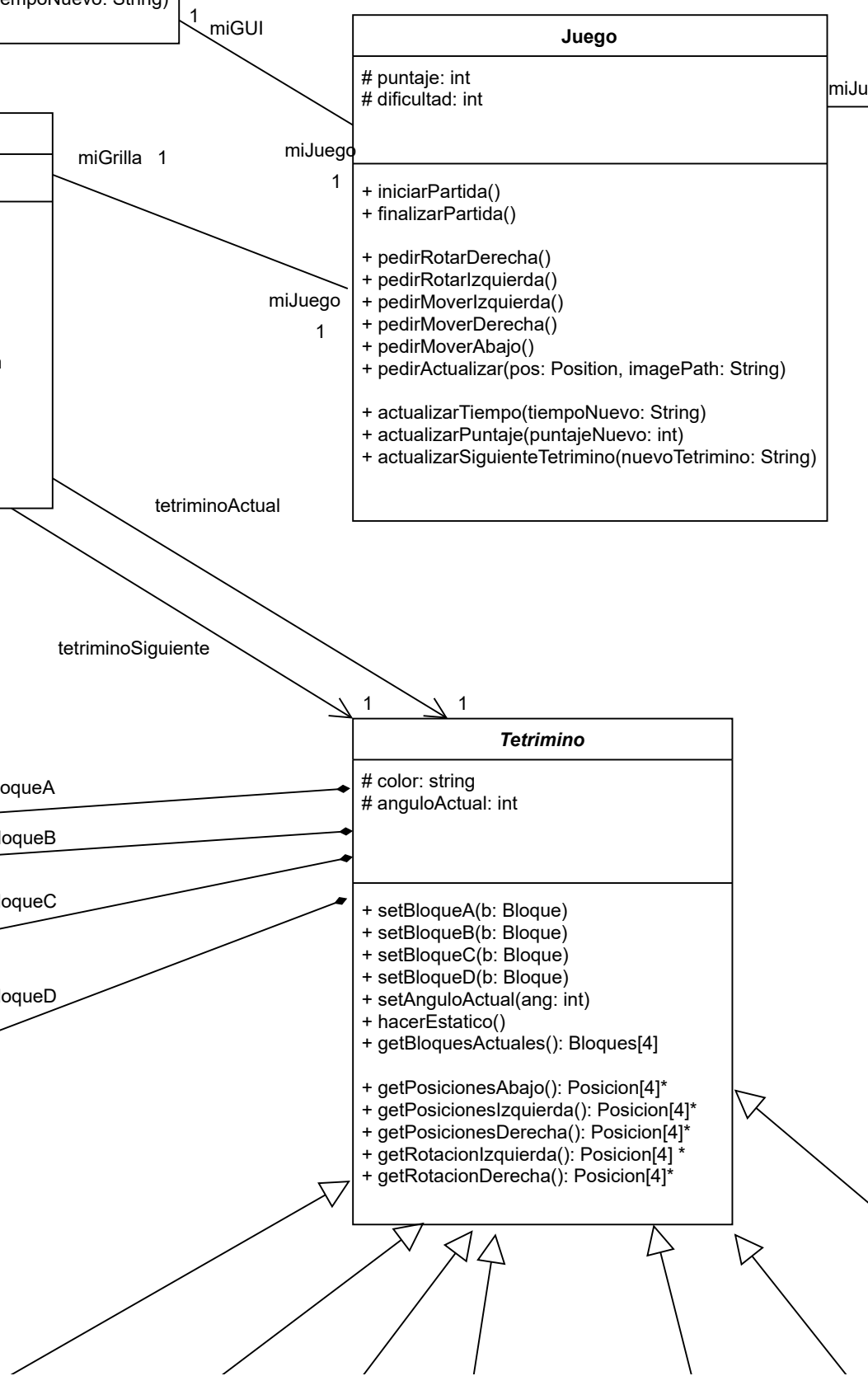
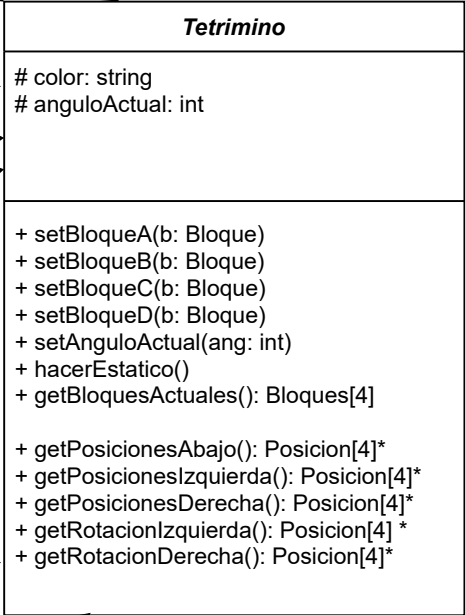
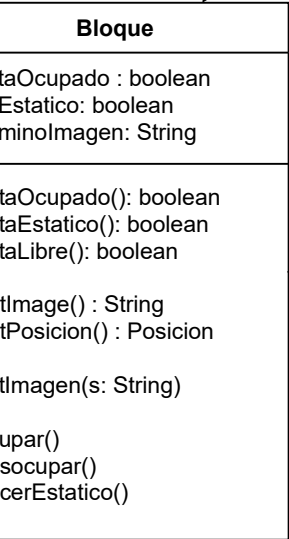
1 pos

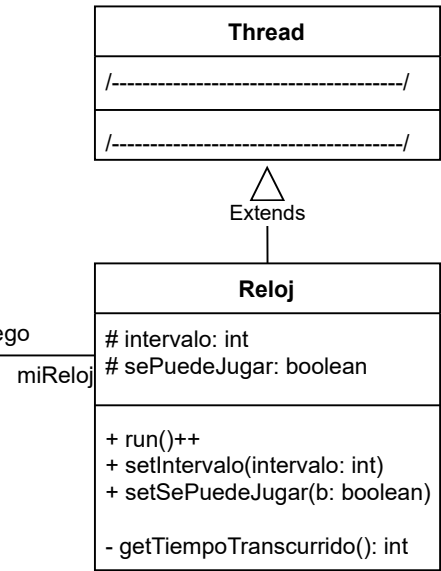


Actualizar es llamada por juego cada vez que un bloque necesita ser actualizado (esto es, si se mueve) va a matrizLabels[i][j] y hace un .setImage(path)



nota: Es una matriz de 21x10





checkearlo las 4 filas correspondientes a las coordenadas Y de los 4 Bloques del Tetrimino, a los cuales accede a través del método getBloquesActuales(). Si es necesario, limpia esas líneas con su operación privada limpiarLinea(), y le notifica a Juego que debe actualizar las imágenes pertinentes. Además le notifica a Juego que debe sumar al puntaje obtenido, y pedirle a la GUI que lo muestre por pantalla con la operación de la clase Juego actualizarPuntaje()

Independientemente de si se tuvieron que limpiar líneas o no, Grilla hace que tetriminoSiguiente pase a ser el tetriminoActual y genera uno nuevo para tetriminoSiguiente. Le notifica a Juego que se debe actualizar la imagen de tetriminoSiguiente y las posiciones de la grilla que deben actualizar su imagen para mostrar el nuevo Tetrimino cayendo.

Sobre el inicio de una partida:

1. Clickeamos Run
2. Se crea la GUI
3. GUI Crea a Juego y se pasa a sí misma como parámetro
4. Juego crea a Grilla y se pasa a sí mismo como parámetro
5. Juego crea a Reloj y se pasa a sí mismo como parámetro (Thread nuevo)
6. Juego empieza el juego (Le da la señal a Grilla y pone a correr al reloj)

TetriminoO
/-----/
+ getPosicionesAbajo(): Posicion[4]++ + getRotacionIzquierda(): Posicion[4]++ + getRotacionDerecha(): Posicion[4]++ + getPosicionesDerecha() : Posicion[4]++ + getPosicionesIzquierda() : Posicion[4]++ - rotacionCero(): Posicion[4] - rotacionNoventa(): Posicion[4] - rotacionCientoOchenta(): Posicion[4] - rotacionDosSetenta(): Posicion[4]

/-----/
+ getPosicion + getRotacion + getRotacion + getPosicion + getPosicion - rotacionCero - rotacionNov - rotacionCien - rotacionDos

