

PRÁCTICA DE PROGRAMACIÓN: TETRIS

1 OBXECTIVO DA PRÁCTICA

Neste exemplo, desenvolvemos un pequeno xogo utilizando de novo a metodoloxía orientada a obxectos.

2 ESPECIFICACIÓNS DOS REQUISITOS

Aínda que o tema do xogo é amplamente coñecido, imos facer un pequeno resumo para aclarar que é o que imos desenvolver:

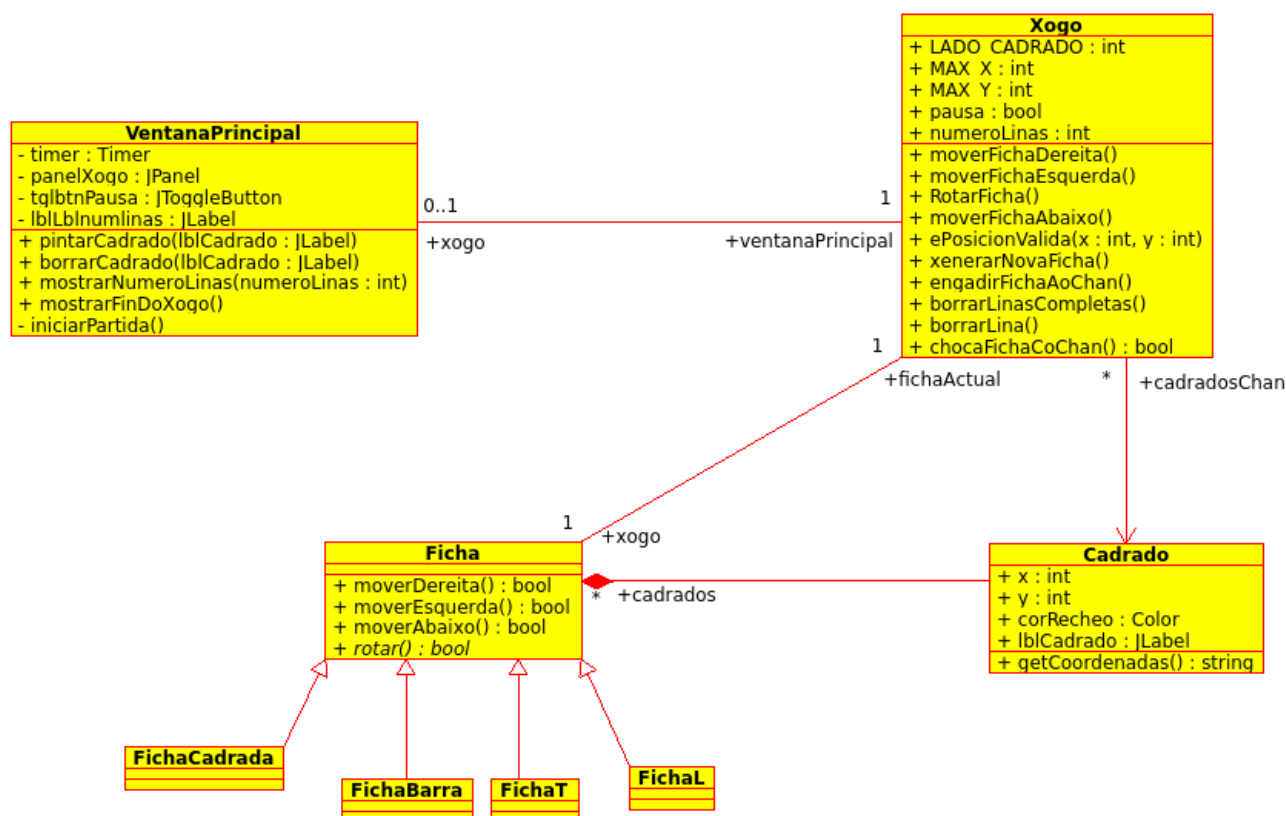
- O xogo consiste en ir colocando unha serie de **fichas** que van saíndo pola parte superior dun panel sobre o chan do mesmo. Estas fichas caen dende a parte superior a unha velocidade que pode variar a medida que avancemos no xogo.
- Cando unha ficha contacta coa parte inferior do panel (o **chan**) ou choca con **cahrados** de fichas anteriores que caeron sobre o chan, a ficha (desfeita en cadrados) pasa a formar tamén parte do mesmo chan.
- Cando no chan conseguimos formar unha fila completamente cuberta de cadrados, entón elimínase esa fila e todos os cadrados que están por riba súa descenden de nivel.
- Para colocar a ficha da mellor forma posible sobre o chan, o xogador pode facer unha serie de accións, como **movela cara a dereita, cara a esquerda, movela cara abaixo e rotala**.
- Cando o nivel do chan chega a un punto tal que unha nova ficha xa non pode saír, remata o xogo. O obxectivo é acadar o maior número de filas eliminadas antes de que o xogo finalice.
- Ao comezo do xogo o usuario poderá escoller o nivel de dificultade iniciar, e este subirá automaticamente cada vez que o usuario acade 5 filas eliminadas.
- Haberá os seguintes tipos de fichas, todas elas formadas por catro cadrados:
 - Ficha cadrada: Formada por dous cadrados enriba e dous cadrados debaixo.
 - Ficha barra: Formada por catro cadrados en fila.
 - Ficha "L": A posición inicial desta ficha son tres cadrados un por riba do outro e un cadrado á dereita do cadrado de abaixo.
 - Ficha "T": A posición inicial desta ficha son tres cadrados en fila e un cadrado debaixo do cadrado do medio.

- Como fichas básicas implementaranse dúas das indicadas. Hai que ter en conta que o método *rotar* dependendo da ficha que se trate terá que ir modificando a posición como se pode ver na seguinte imaxe:

POSICIÓN 0	POSICIÓN 1	POSICIÓN 2	POSICIÓN 3
<div> <div>0</div> <div>1</div> <div>23</div> </div>	<div> <div>3</div> <div>012</div> </div>	<div> <div>32</div> <div>1</div> <div>0</div> </div>	<div> <div>210</div> <div>3</div> </div>
POSICIÓN 0	POSICIÓN 1		
<div> <div>0</div> <div>1</div> <div>2</div> <div>3</div> </div>	<div> <div>0123</div> </div>		
POSICIÓN 0	POSICIÓN 1	POSICIÓN 2	POSICIÓN 3
<div> <div>0</div> <div>13</div> <div>2</div> </div>	<div> <div>012</div> <div>3</div> </div>	<div> <div>0</div> <div>31</div> <div>2</div> </div>	<div> <div>3</div> <div>012</div> </div>

3 DIAGRAMA DE CLASES

Facendo unha análise dos requisitos explicados, poderíamos chegar a un diagrama de clases como o que se ve na imaxe.



Imos explicar brevemente o significado do diagrama e as clases que contén:

- A clase **VentanaPrincipal** implementa a ventá gráfica do xogo. Como atributos terá o panel do xogo no que aparecerán as fichas, o botón que podemos usar para pausar o xogo e a etiqueta que mostra o número de liñas conseguidas. Tamén manterá unha referencia a un timer para mover as fichas cara abaixo de forma automática. Inclúe métodos para pintar un cadrado no panel ou borrarlo, mostrar o número de liñas eliminadas que se encargará de acelerar o timer cada 5 liñas, mostrar unha mensaxe no final do xogo e iniciar unha nova partida. No momento en que se inicia unha nova partida, creárase un obxecto da clase **Xogo** que será ao que se chame cada vez que se queira executar algunha acción durante a partida.
- A clase **Xogo** mantén o estado dunha partida e implementa as funcións dispoñibles dentro da mesma. Define unha serie de constantes que definen o tamaño dos cadrados que forman as fichas e o panel do xogo, o atributo pausa que indica se o xogo está ou non pausado e o número de liñas eliminadas na partida. Un obxecto **Xogo** sempre está asociado a un obxecto **Ficha** que se corresponde coa ficha actual, que é a única que se pode mover e rotar, usando os métodos *moverFichaDereita*, *moverFichaEsquerda*, *moverFichaAbaixo* e *rotarFicha*. A clase

Xogo tamén implementa unha serie de métodos para controlar a creación e o movemento das fichas e a eliminación das filas de cadrados completas no chan.

- A clase **Ficha** implementa as operacións dunha ficha, e será estendida polas distintas clases que implementan os tipos de fichas existentes. Veremos que as operacións de mover á dereita, esquerda e abaixo son idénticas para todos os tipos de fichas, mentres que o método *rotar* variará dependendo do tipo de ficha do que se trate, polo que será implementado de forma diferente en cada clase de ficha.
- Por último, a clase **Cadrado** implementa o elemento básico que forma unha ficha, e que logo pasarán a formar parte do *chan* do xogo cando a ficha toque ese chan. O cadrado ten como atributos as coordenadas x e y nas que se sitúa dentro do panel, a cor de recheo do cadrado e unha referencia ao compoñente gráfico que fai que se visualice o cadrado (neste caso, un compoñente de *Swing* chamado *JLabel*).

Haberá que clasificar as clases do programa en diferentes paquetes:

- Paquete **iu** (interface de usuario): No que debe estar a clase *VentanaPrincipal*.
- Paquete **modelo**: No que deben estar as clases *Xogo*, *Ficha* e *Cadrado*.

4 PARTE OPCIONAL

- Implementar outros tipos de fichas.
- Se o usuario chega a un nivel determinado, o xogo engadirá cada certo tempo unha fila con un certo número cadrados colocados aleatoriamente no nivel máis baixo do chan, que fará subir un nivel a todas os cadrados que haxa nese momento no chan.
- Reproducir un son cada vez que o usuario elimina unha fila.