Existe una clase principal llamada Bloque. Esta clase contiene:

1. El evento detener
2. El evento mover
3. Un conjunto de cuadros
4. Un conjunto de colores de fondo
5. Un conjunto de colores de frente
6. Un tipo de bloque (J-L-Z-SQUARE-T-S-LINE)
7. Un tamaño para el cuadrito
8. Un tipo de rotación
9. Un estado para detenido

La clase bloque tiene dos constructores:

1. Un constructor que recibe como parámetros la ubicación y el tipo de bloque: el constructor inicializa la ubicación, el tamaño para el cuadrito, el conjunto de 4 cuadros, el tipo y crea los cuadros.
2. Un constructor que recibe como parámetros la ubicación: el constructor inicializa la ubicación, el tamaño para los cuadros, el conjunto de 4 cuadros y ejecuta una función aleatoria que obtiene un tipo de bloque y crea los cuadros

La clase tiene los siguientes métodos:

1. CrearCuadros: según el tipo de bloque crea las ubicaciones para los diferentes cuadros
2. Abajo: En caso que se pueda bajar, oculte, por cada cuadrito crear una nueva ubicación, a la ubicación del bloque asignarle la ubicación del primer cuadrito, mostrar, si no hay evento de movimientos crear un evento de movimiento devolver verdadero, en caso contrario detener devolver false.
3. Derecha: en caso que se pueda ir a la derecha, oculte, por cada cuadrito crear una nueva ubicación, a la ubicación del bloque asignarle la ubicación del primer cuadrito, mostrar, devolver verdadero, en caso contrario devolver falso.
4. Izquierda: en caso que se pueda ir a la izquierda, oculte, por cada cuadrito crear una nueva ubicación, a la ubicación del bloque asignarle la ubicación del primer cuadrito, mostrar, devolver verdadero, en caso contrario devolver falso.
5. Rotar: Según el tipo de Bloque y la rotación actual crear la nueva ubicación de cada cuadrito, en caso que no se pueda rotar devuelve a la posición anterior.
6. Mostrar: muestra cada cuadrito
7. Ocultar: oculta cada cuadrito
8. Devolver: devuelve a la posición anterior

La clase tiene las siguientes propiedades:

1. Top
2. RotacionActual
3. Cuadrito 1
4. Cuadrito2
5. Cuadrito3
6. Cuadrito4
7. TipoBloque

Existe una clase CampoJuego que contiene:

1. Un color de BackGround
2. Un punto para Display
3. Un tamaño para cuadrito
4. Una posición para la parte inferior
5. Una posición para la parte izquierda
6. Una posición para la parte derecha
7. Una posición para la parte superior
8. Un número de cuadros por línea

La clase tiene un constructor que llama al método redibujar

La clase tiene los siguientes métodos:

1. EstaVacio: según la dirección, por cada cuadrito pregunta si en la matriz de cuadros total existe la nueva posición, en caso que exista devuelve false
2. DetenerCuadrito: por cada cuadrito en el bloque, si no existe la posición del cuadrito en la matriz de cuadros total agrega el nuevo cuadrito.
3. ChequearLineas: por cada línea de la matriz, en caso que la línea este llena, baja todas las líneas y por cada línea baja la posición de sus cuadros.
4. Redibujar: por cada línea en la matriz de cuadros totales muestra los cuadros

Debe existir una clase Bloque básica que contiene:

1. Un evento detener: ocurre cuando un bloque no puede continuar bajando.
2. Un evento mover: ocurre cuando se mueve un bloque a la izquierda, derecha, abajo o arriba para rotar.
3. Una posición inicial.
4. Un conjunto de 4 cuadros que representan un bloque
5. Un color de bloque
6. Un tipo de bloque
7. Un tipo de movimiento (izquierda, derecha, abajo o arriba)
8. Un tipo de rotación
9. Un estado que indica si está o no detenido
10. Una función para desplazar a la izquierda (General para todo tipo de bloque)
11. Una función para desplazar a la derecha (General para todo tipo de bloque)
12. Una función para desplazar abajo (General para todo tipo de bloque)
13. Una función encargada de crear las ubicaciones de los cuadros según el tipo de bloque
14. Una función para rotar según el tipo de bloque
15. Una función para mostrar el bloque
16. Una función para ocultar el bloque
17. Una función Aleatorio que genera un número que indica el tipo de bloque que se va a dibujar

Debe existir una matriz con todas las posiciones que puede ocupar un cuadro en el tablero (x,y) y llenarlas de 0, si un cuadro está ocupando una posición el valor que toma es 1. El valor 1 se da cuando un cuadro ya llegó al fin del movimiento

Los valores 0 y 1 son tenidos en cuenta para decidir si es posible hacer un movimiento o si una línea de cuadros ya está llena y como debe ser destruida esa línea.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0,0 | 10,0 | 20,0 | 30,0 | 40,0 | 50,0 | 60,0 | 70,0 | 80,0 | 90,0 | 10,0 | 110,0 | 120,0 | 130,0 | 140,0 | 150,0 |
| 0,10 | 0 | 0 | 0 | 0 |  |  |  |  |  | 0 | 0 | 0 | 0 | 0 | 0 |
| 0,20 | 0 |  | 0 |  |  |  |  |  |  |  |  |  |  |  | 0 |
| 0,30 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  | 0 |
| 0,40 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  | 0 |
| 0,50 | 0 | 0 |  |  |  |  |  |  |  |  |  |  |  |  | 0 |
| 0,60 | 0 | 0 | 0 | 0 | 0 |  |  |  |  |  |  |  |  |  | 0 |
| 0,70 |  |  | 0 |  | 0 | 0 |  |  |  |  |  |  |  |  | 0 |
| 0,80 |  |  | 0 |  |  |  |  |  |  |  |  |  |  |  | 0 |
| 0,90 |  |  | 0 | 0 |  |  |  |  |  |  |  |  |  |  | 0 |
| 0,100 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  | 0 |
| 0,110 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  | 0 |
| 0,120 | 0 |  |  |  |  |  |  |  | 1 |  |  |  |  |  | 0 |
| 0,130 | 0 |  |  |  |  |  |  |  | 1 |  |  |  |  |  | 0 |
| 0,140 | 0 |  |  |  |  |  | 1 | 1 | 1 |  |  |  |  |  | 0 |
| 0,150 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

Debe existir una clase Cuadro que contiene:

1. Un tamaño para el cuadro
2. Un color
3. Una posición
4. Una función que muestra el cuadro
5. Una función que oculta el cuadro
6. Una función que baja el cuadro (aumenta la posición Y)
7. Una función que mueve el cuadro a la izquierda
8. Una función que mueve el cuadro a la derecha

Debe existir una Clase propia de la aplicación que contiene:

1. Una función que consulta si un cuadro puede desplazarse a la izquierda, a la derecha, abajo o arriba para rotar con base en la matriz de posiciones y en la dirección que se desee.
2. Una función que ingrese un cuadrado a la matriz.
3. Una función que elimine una fila llena de la matriz
4. Una función que vuelve a dibujar todos los cuadros