BaseCase Programming Test

You have 9hrs to implement the following game. In order of importance, we are interested in:

1. full, bug-free implementation of the spec

2. elegance of the code

3. time to completion

Note on the 'elegance of the code' - this includes the following:

1. Clear, sensible abstractions

2. Short and precise code (though - not so terse as to inhibit readibility)

3. No redundancies

When you are done, please send us the full source code and appropriate build instructions. It is important to reply to the same email that contained this test.

Note that in the event that you do not pass this test, we will not be able to provide any feedback on the reason.

Tetris

Implement a simple 'text-mode' version of the 'Tetris' game, following the specification below. Please don't go outside of the specification, adding features not explicitly requested. You are free to use a language of your choice. There are 5 pieces In this version of Tetris:

```
****
```

```
*
*
**
```

```
 *
 *
**
```

```
 *
**
*
```

```
**
**
```

and they fall down a 20x20 tetris board:

```
*                     *
*                     *
*                     *
*                     *
*                     *
*                     *
*                     *
*                     *
*                     *
*                     *
*                     *
*                     *
*                     *
*                     *
*                     *
*                     *
*                     *
*                     *
*                     *
**********************
```

The game starts with a random piece appearing at the top of the board. The user is then prompted to make a move:

- a (return): move piece left

- d (return): move piece right

- w (return): rotate piece counter clockwise

- s (return): rotate piece clockwise

If the move the user selects is valid, then it is executed and the screen redrawn (you can use printf()/ cout / System.out.println(), etc to redraw the entire board). If it is not valid then they are again prompted to enter a valid move.

Note that the game only updates after the user has entered a valid action.

A valid move is defined thus: The piece is altered as per the user's input, and then displaced by 1 row downwards. If the piece, drawn at it's new location, is not outside the bounds of the board, and does not overlap any pieces that previously fell, then it is a valid move.

If the piece's new position is such that it allows no valid move, then a new piece appears along the top of the board, randomly positioned along the x-axis. If this new piece happens offer no valid move, then the game is over and the program exits.

If you feel some part of this spec is unclear/contradictory in some detail, please resolve it using your best judgement, and make a note explaining your decision.

Good Luck!