Homework 4: Tetris

Submission:

All submissions should be done via git. Refer to the git setup, and submission documents for the correct procedure.

The root directory of your repository should contain your README file, and your Android Studio project directory.

Overview:

“If Tetris has taught me anything, it's that errors pile up and accomplishments disappear.”
– Some guy on Reddit

Using a framework provided you will be implementing a simple game of Tetris.

For those who have never played tetris, definitely try it out. Below is a link to a free online tetris game. Use the “up” arrow key to rotate the blocks.

https://www.freetetris.org/game.php

On the following page is a screenshot of a version made by the TA.

For previous assignments, layout and quality of appearance have not been major factors in grading. However, appearance will be considered for this assignment. Your app should look at least as polished as my example app.
API:

I have provided several framework classes for your use. These classes manage the internal logic of many parts of a Tetris game. You are not required to use these files.

TCell: A class that represents a single square. Each TCell has a color and (x,y) coordinates.

TGrid: A class that creates a grid of cells. Use this to represent the main grid of cells in your game. This class also provides several utility functions that help you to detect and delete full rows of cells.

Tetromino: A class that represents a Tetris Tetromino. This class can be inserted into a TGrid, moved, and rotated. All of Tetromino's movement functions return a boolean value depending on whether the movement succeeded or not.

TetrominoBuilder: This class returns a variety of fully constructed Tetrominos.

Specifications:

Elements

Your app should contain the following elements:

- A grid 20 tall and 10 wide in which to play Tetris
  - There should be three invisible rows above the main grid. This is where you should place new Tetrominoes (so the full grid should be 23 tall).
  - Blocks should be square. Make your visible grid exactly twice as tall as it is wide.
- Cells that are displayed on the screen should have a border.
- A tiny window that displays the next Tetromino that will appear.
- Text that displays the current number of rows that have been cleared
- Text that displays the current score
  - Every time a row is cleared, add a number of points equal to the current level to the score. At level 1 add one point for each row, at level 2 two points, etc.
- Text that displays the current level
  - Start at level 1.
  - Every time five rows have been cleared increase the level by one.
  - Every time the level is increased by one, decrease the time between each block movement by 20%
- A reset button (starts new game)
- An option in the menu bar that allows the user to increase the level by 1.
- Controls: left, right, rotate left, rotate right, and down.
  - It should be obvious to me by looking at your app which controls are which
  - left/right shift the tetromino to the left or right respectively
  - rotate should rotate the tetromino left or right.
○ down should cause the tetromino to drop immediately into place.
  • Notify the user when the game has ended.

Other Notes

At the beginning of your game, the time between each movement should be roughly one second. As the level increases this time will decrease.

When you insert new Tetrominoes, insert them into position (4,0).

The game is over when you attempt to insert a Tetromino and cannot do so.

Extra Credit

  • High score board; Must be implemented as a ListView or RecyclerView a new activity. User should be taken to that activity when the game ends (and not when it ends because the user pressed reset!).

Hints:

http://tetris.wikia.com/wiki/Tetris_Wiki

Helper files

The coordinate system of a TGrid object starts in the upper left corner. The positive X axis points to the right, the positive Y axis points down.

Create a custom class (extending View) to display the game board for the Tetris game. Then all you need to do is override its onDraw() function.

This custom class should take a TGrid object (which handles game logic) and render its contents to the screen.

If you design your class well, you should be able to reuse it to render the next Tetromino that should be displayed.

This code snippet will draw a red square.
n the above example I draw a square with a fixed size. For this assignment, it may be beneficial to draw your squares as a fraction of the view's width and height.

Call the .invalidate() function on a View to cause it to redraw itself.

Don't forget to check if multiple rows can be deleted.