Assignment 3 – A Complete 2D Tetris (8%+2% Bonus)

Due: 11:59pm, Sunday, 16 October 2016

Q1(8%): Complete your Tetris program (with your Assignment 2 submission) to include the following additional steps and functionality:

- Add the following constants which could be adjusted as needed in the indicated ranges:
  - *M* – scoring factor (range: 1-10).
  - *N* – number of rows required for each *Level* of difficulty (range: 20-50).
  - *S* – speed factor (range: 0.1-1.0).
- When any horizontal row of squares *R* has no hole, i.e. all the squares in *R* are parts of some shapes with colors, *R* is removed and all the rows above *R* move one square down, *Lines = Lines + 1; Score = Score + Level x M*.
- If the number of removed rows in the current *Level* reaches *N, Level = Level + 1*, the falling speed *FS = FS x (1 + Level x S).*
- When a new shape has no space to fall, i.e. existing shapes in “Main area” pile up to near the top, the game terminates.
- If the cursor is inside the falling shape *F* (in PAUSE mode), *F* will be changed to one of the shapes different from *F* and that currently inside “Next shape”, *Score = Score - Level x M.* You should use “Point-Inside-Polygon” test algorithm to detect the cursor.
- Create a user-friendly interface so that various parameters could be adjusted, using GUI widgets of your choice (e.g. a slider for *M*), to suit different user groups:
  - Constants *M, N,* and *S* are individually adjustable.
  - The width and height of “Main area” can be adjusted (beyond 10 x 20 squares).
  - The size of the square is adjustable (e.g. enlarged for elderly players).

Note: The above updating functions are very simple (by multiplying a constant). You may introduce more realistic and complex functions and may vary the ranges of *M, N,* and *S* to reflect more interesting playing experiences. Describe your changes, as well as the general implementation method, in the comments on top of the source code.

Q2(2% Bonus): New shapes with different square compositions (up to 3 squares in each new shape) and colors may be added, such as those in the following figure. Your user interface will allow the player to compose squares to make new shapes, or will display a fixed set of shapes for the player to select and add.

Submit your Java program in one file on WebCT before the due time. Late submissions should be submitted to the TA through email rather than through WebCT, and will be penalized with a 2% deduction for every 1-24 hour delay.