DUE DATE:

ALL recitations: 9:00 PM on 12/09

The week of 11/28 - 12/3 recitations and TA office hours will run as usual.

The week of 12/5 - 12/9 recitations in Baldy 21 will run as office hours (attend any one you’d like), but there will be NO regular TA office hours in Davis 302.

Preliminaries

This lab is the third of three labs (lab 8, lab 10 and lab 11) that in the end will have you build a single-player matching game like Candy Crush.

For this lab you have an option: continue with the code you wrote for lab 10, or use our lab 10 solution code. Note: even if you did not complete lab 10 you can attempt lab 11. The lab 10 solution code is available in the repository.

Keep in mind that any work you submit must be your own. Submitting work done by someone else as your own is academically dishonest, and will result in immediate failure in the course. We will use software tools to detect inappropriate collaboration.

Your TAs will give you an overview of the lab and hints of how to proceed in recitation. You should expect to have to put in time outside of recitation to finish this lab. We therefore recommend that you bring your laptop to recitation so you can more easily continue to work outside of Baldy 21.

During office hours TAs will give priority to students who have attended recitation. It is not acceptable to skip recitation and then expect one-on-one assistance in office hours. Some aspects of this lab will ONLY be discussed in recitation, and TAs will NOT answer basic questions about these topics during office hours.

Because this lab gives you a great deal of freedom in how you write the code there is NO AUTOMATED GRADING. When you submit to Web-CAT the only score you will see is your early bonus or late penalty. The functionality of this lab will be manually graded by the TAs.
Overview

For this lab you will have various options of functionality to build. It is up to you how much or how little you do, but you must complete ALL the required functionality before ANY optional functionality will be graded. Completing only the required functionality will NOT earn you full credit. You can earn extra credit on this lab. Full credit on the lab is 100 points. You can build this lab either from your lab 10 solution or the lab 10 solution we are providing.

Required functionality [80 points total - remember 100 points are needed for full credit]:

[20 points/REQUIRED] Initial configuration
Display a 5 x 5 grid of colored tiles. The initial board must satisfy the following constraints:
- it must not have any matches (10 points)
- it must have at least one legal move (10 points)

[20 points/REQUIRED] Legal move
Adjacent tiles can be interchanged only if the exchange results in a match. Non-adjacent tiles cannot be interchanged. Adjacent tiles whose exchange does not result in a match cannot be interchanged.

[20 points/REQUIRED] Match response
A match is defined as at least three adjacent tiles of the same color, appearing in a straight line either horizontally or vertically.

After a match is identified:
1. remove the matched tiles,
2. drop down the tiles above the ones removed, filling in with random tiles at the top
Repeat steps 1 and 2 as long as there are matches on the board.

[20 points/REQUIRED] Game end
End game when no more turns are available. Use System.out.println to print out the score (if scoring is not implemented, print out “Game over” instead) before exiting the game using System.exit(0).

Optional functionality [Maximum 90 points available]:

OPTIONAL:
[20 points] Move hints
Highlight one possible move at start of each turn. Remember that if no turn exists then the game must end.
OPTIONAL:
Scoring (submit an implementation for at most one)
[10 points] Basic scoring: 3 points for a match, regardless of size.

[20 points] Intermediate scoring: Each match scored according to the number of tiles it contains. A match of length 3 scores 3 points, whereas a match of length 4 scores 4 points, etc.

[30 points] Advanced scoring: Each matched region scored according to the following formula:

The size of matched region determines score, according to this formula (where x is the size of the matched region):

score = 3 + (x-3)*(x-3)

Matched region - definition
A matched region consists of all contiguous matches of the same color. Adjacent tiles of the same color not independently part of a match are NOT part of the matched region. [I will give examples.]

OPTIONAL:
[20 points] High score (you may need to do some independent research to learn how to do this)
Persist the high score to a file. This is should be a single integer - the highest score that’s been achieved. The high score must be loaded from a file named “highScore.txt” at start of game. The high score must be displayed in the UI together with the current game’s score. At end of a game the score is compared to the high score. If the new score is higher than the old high score, it becomes the high score and is saved to the highScore.txt file.

OPTIONAL:
[20 points] Levels
Introduce levels for the game. Each level has a board that is one row and one column larger than the last. The first level has a 5x5 board, the second level has a 6x6 board, and so on. In general Level #n has (n+4)*(n+4) board.

Each level has a point goal for proceeding to the next level. The point goal for each level is 10 times the # rows. Thus level 1 has a goal of 50 points, level 2 has a goal of 60 points, etc.

Clarification: THE GOAL IS THE NUMBER OF POINTS THAT MUST BE EARNED ON THAT LEVEL (NOT THE OVERALL GAME SCORE).

The stopping condition is the same as before: when there are no more moves available the game ends.

Submitting your project to Web-CAT
Make sure you submit your work on time; due dates are listed at the beginning of this lab description. There is NO automated grading for this lab. You may submit as many times as you wish. Your last submission is the one that counts (so consider carefully whether you want to make any late submissions, as the late penalty is 20 points per day or portion thereof late).