Tetris Design Section Mini-Assignment

Due Date: Monday, November 4th, at 11:59PM

Note: You must email your section TAs a PDF of your mini assignment by the due date listed above. Submissions after the deadline or by any other format will not be accepted.

Instructions

- Read the assignment handout carefully and play around with the Tetris demo before you delve into this mini-assignment. We recommend completing this mini-assignment before you begin coding.
- A reminder that the diagrams can be drawn in the digital program of your choice, but they cannot be hand-drawn.
- Bring a printed copy of your answers to your design section so your design section TAs can facilitate a productive and collaborative conversation.

Questions

1. What data structure will you use to represent the board? What type of object will this data structure hold?

2. An important logical step in this program is clearing completed lines off the board. Write pseudocode for how you will check for and clear completed rows. This involves both “graphical clearing” i.e. hiding the cleared squares from the screen and moving the rows above down as well as “logical clearing” i.e. updating the board to know about the new arrangement of squares. The steps you write should clearly outline all cases you would cover to properly clear lines.

3. Draw the following diagrams:
   a. Create a simple logical containment diagram for your program. All classes used in your proposed design should be modelled in this diagram, including JavaFX classes. Think about the purpose of each class as you design your program and be prepared to discuss it.
   b. If your design requires it, create an inheritance diagram for your program, showing which classes extend other classes and which interfaces (if any) they implement.

4. An important part of this project is checking move validity. As the current piece is moving/falling - you will want to make sure it does not fall off the edge of the board or onto an already fallen piece. Think about how you would keep track of and check for move validity? (It will be useful to consider the tradeoffs of whether the board, the piece, or the individual components of the piece should keep track of this.)
5. A plan for coding Tetris incrementally. Tetris is a large project, and should be coded in small, manageable steps. See earlier handouts for examples of a plan for coding incrementally -- we highly recommend actually writing this out before you begin coding!

CS Responsibility

Autonomous driving is one of the most encouraging applications of machine learning that we have today. However, the promise of this technology comes with a lot of ethical concerns that are related to moral decision making. Read these articles to get a sense of what sort of ethical dilemmas that autonomous driving will have to address, and what implications these concerns may have.

This first article discusses 4 of the main concerns with autonomous vehicles: https://news.stanford.edu/2017/05/22/stanford-scholars-researchers-discuss-key-ethical-questions-self-driving-cars-present/
The second article discusses the trolley problem in relation to autonomous vehicles: https://www.wired.com/story/trolley-problem-teach-self-driving-car-engineers/

Upon reading the articles:

1. In what ways does the trolley problem apply to the problems concerned with autonomous vehicles?
2. Which of the four ethical concerns in the first article most concerns you most? Why?