Assignment #1 (30 marks)

Written part due: February 16th, Monday, in class.
Programming part due: February 16th, Monday, at 11:45 pm.

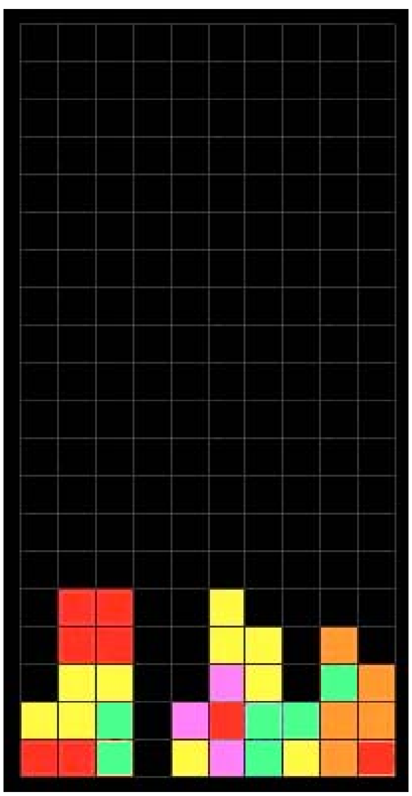
On the cover page of your homework, write and sign the following statement: "I have read and understood the policy concerning collaboration on homework and lab assignments." Without such a signed statement, your work will not be marked. A sample cover page can be found under the Files folder in Canvas.

Problem 1 (25 marks): Tetris meet Falling Fruits!

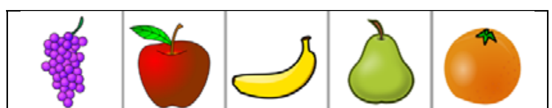
You will implement a simple interactive game that combines some features from Tetris and Falling Fruits. Any visual flare that you wish to add to the appearance of your game will be judged by the grader and may be credited at his discretion.

The Falling Fruits game is similar to Tetris. (An example Falling Fruits game can be found at http://www.wordgames.com/search.html?query=falling+fruits).

Your game window consists of a 20x10 square grid of appropriate size, e.g., so that the window will fit in the screen comfortably. Different fruits are represented by squares of different colors. You program should have five different fruits with color coding specified in figure (b). Each time, four fruits fall from the top of the screen. (Note, the online Falling Fruits game is different and falls only three fruits at a time.) As indicated in the figure (c), these fruits are arranged in a I, S or L shaped Tetris tile, which has a pivot of rotation indicated by a black dot. You are advised to complete this problem in several steps.

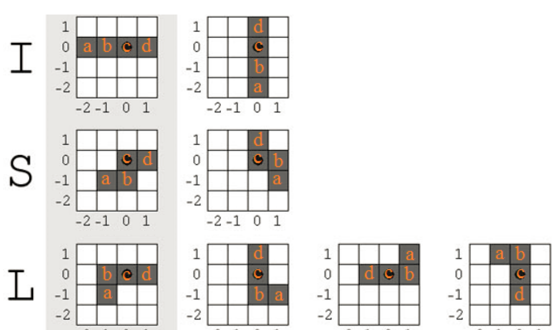


(a) The screen of the falling fruit game.



Grape	Apple	Banana	Pear	Orange
Purple	Red	Yellow	Green	Orange

(b) To simplify the task, we use squares of different colors representing different fruits. This table shows the color coding of fruits.



(c) The four fruits are arranged in a row or column.

(a) [6 marks] Tile and grid rendering and tile downward movement

Set up the game window with grid lines. At each time, randomly select a Tetris tile consisting of four fruits and drop it from the top of the game window. The starting position and orientation are chosen randomly. You can control the speed of its movement to suit your game playing. Movement of the tiles will be aligned with the grids and at uniform speed. For this step, the tiles can drop straight through the bottom. After one array disappears, a new array is dropped.

(b) [3 marks] Stack-up

In this step, the tiles will stack up on top of each other and the bottom of the game window will offer ground support.

(c) [9 marks] Key stroke interaction and tile movements

The four arrow keys will be used to move the dropping fruit array. A pressing of the “up” key rotates the tile *counterclockwise* about its pivot, 90° at a time. The “left” and “right” key presses result in lateral movements of the array, one grid at a time. The “down” key accelerates the downward movement. At no time should you allow a tile to collide with any existing tiles or the border of the game window. The “space” key will shuffle the fruits, which moves the fruit within the tile according to the following order: a->b, b->c, c->d, d->a.

(d) [7 marks] Additional game logic

1) When three same fruits are in a row or column, they will be removed and the tiles above them will be moved down. 2) When a row is completely filled, it is removed and the tiles above it will be moved one row down. Game terminates when a new tile piece cannot be fit within the game window. Press ‘q’ to quit and ‘r’ to restart. Pressing any of the arrow keys should not slow down the downward movement of a tile.

Note that the above steps build on top of each other, in order. You need not submit individual programs to correspond to these steps. If you can implement all the required parts, a single, complete program is sufficient. ***A skeleton code*** for Tetris will be provided.

<u>**Submission**</u>: All source codes, a ***Makefile*** to make the executable called ***FruitTetris*** (the make command should be make), and a ***README*** file that documents any steps not completed, additional features, and any extra instructions for your TA.

Problem 2 (5 marks): Rigid-body transformations

Let us recall that the general form of a 3D transformation matrix M in homogeneous coordinates is

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Assume that the upper 3 by 3 submatrix R of M is orthonormal, i.e., $R^T = R^{-1}$.

1. [2 marks] What is the inverse of M? Note that you should not use brute force or a package such as Maple or Matlab to answer this question.

2. [3 marks] Prove that the transformation M preserves both lengths and angles in 3D. Note here that when we talk about the angle between two vectors, the order in which the two vectors are given would be irrelevant.