Assignment #2 (30 marks)

Programming part due: Mar 11th, Wednesday, at 11:45 pm.

Problem 1 (30 marks): Playing Tetris with a robotic arm

You will upgrade the Tetris-FallingFruits game in assignment 1 into 3D, and control a robot arm to play this game. Check out the sample code for robot arm in:

http://www.cs.unm.edu/~angel/BOOK/INTERACTIVE_COMPUTER_GRAPHICS/SIXTH_EDITION/CODE/CHAPTER08/
and look into example1.cpp and associated shaders.

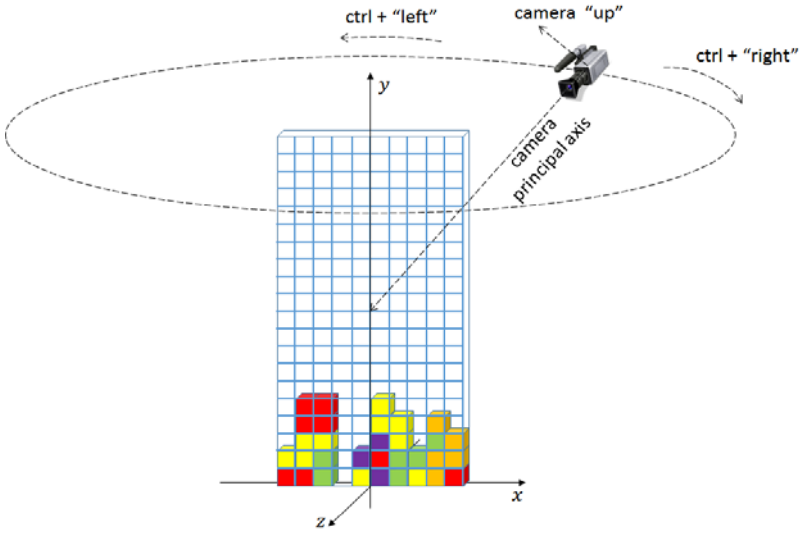
(a) [5 marks] Upgrading Assignment 1 to 3D

Upgrade your Tetris-FallingFruits game to 3D by turning each fruit from a 2D square to a 3D cuboid. The centers of all fruit cells are constrained in a 2D plane, e.g. $z = 0$. You can keep the same game logical at this stage.

In case you feel uncomfortable with your implementation of Assignment 1, you can begin with the TA's sample code for the Tetris game.

(b) [5 marks] Viewpoint Changes

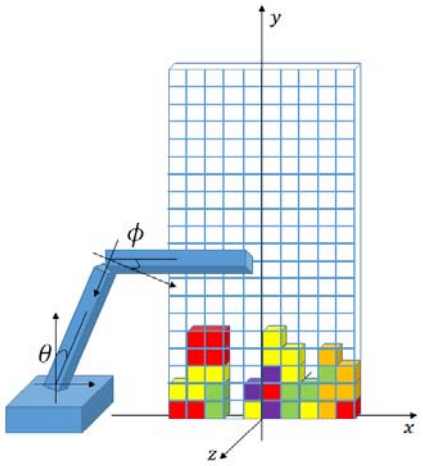
The "ctrl" key will be combined with the "left" and "right" key to control the camera's movement on a circular orbit that is parallel to the x-z plane with a center on the y axis. The camera always looks at the center of the Tetris window. Its "up" direction lies in the plane spanned by the y axis and the camera's principal axis – the line connecting the camera center and the "Lookat" point.



(c) [5 marks] Control of the Robot Arm

Put the robot arm in the 3D space of Tetris tiles. The central axes of both lower and upper arms will be limited in the same plane as fruit cuboids. In this way, the lower arm is controlled by a rotation angle θ, and the upper arm is controlled by an angle ϕ. Use the "a" (or "d") key to increase (or decrease) the angle θ, and the "w" (or "s") key to increase (or

decrease) the angle ϕ. Set the arm length appropriately, such that it can reach every location of the Tetris window.



(d) [5 marks] Game Logic

New Tetris tiles only appear at the tip of the robot arm. Rotate the robot arm to put it at desired locations. At this stage, it is OK to ignore the collision between the new tile and existing ones. Use the "space" key to drop a Tetris tile from the robot arm. The tile's position should align with the grid, i.e. snapped to the nearest grid position. Once a tile is dropped, it can either float at the location it is dropped, or keep dropping downward until it hits some other tiles (like in Assignment 1).

In this assignment, you don't have to shuffle the fruits, since the "space" key is used for dropping. If you want to keep this feature, you might use ctrl + "space" for shuffling. The "up" key is still reserved to rotate a tile. The "left" and "right" keys are disabled. Press 'q' to quit and 'r' to restart. Eliminating full rows and three fruits with the same color is optional and will not be judged since it is evaluated in Assignment 1.

(e) [5 marks] Collision Detection

The new Tetris tile can only be dropped at places without collision with existing tiles. When the tile is attached to the robot arm, we allow collision, but will highlight collision cases by turning the tile to grey color (so that the player knows it cannot be dropped).

(f) [3 marks] Bonus Points

The placement of each tile is limited in certain amount of seconds. Display the remaining seconds (as a text message) for each tile on the top of the window.

When the time is out, the tile on the arm will be replaced by a new one, or be automatically dropped at the current location. This game logical will not be graded. The bonus points are for the display of the timing.

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. ***Skeleton codes*** for Tetris and Robot Arm are provided.

Submission: All source codes, a ***Makefile*** to make the executable called ***FruitTetris3D*** (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.