

CSE4204 | Section B | Computer Graphics Lab | Assignment – 3

[10 marks]:

Create a crate (box) with a lid. Both the crate and lid have the same coordinates; only lid is the scaled (skewed) version of the crate. You have freedom to choose colors. Attach two separate shader programs for the crate and the lid. Apply perspective projection on them. For each left and right arrow key pressing, the crate along with the lid (crate + lid) will rotate positive and negative degree along Y axis respectively. For each up and down arrow key pressing, only the lid will open and close respectively.

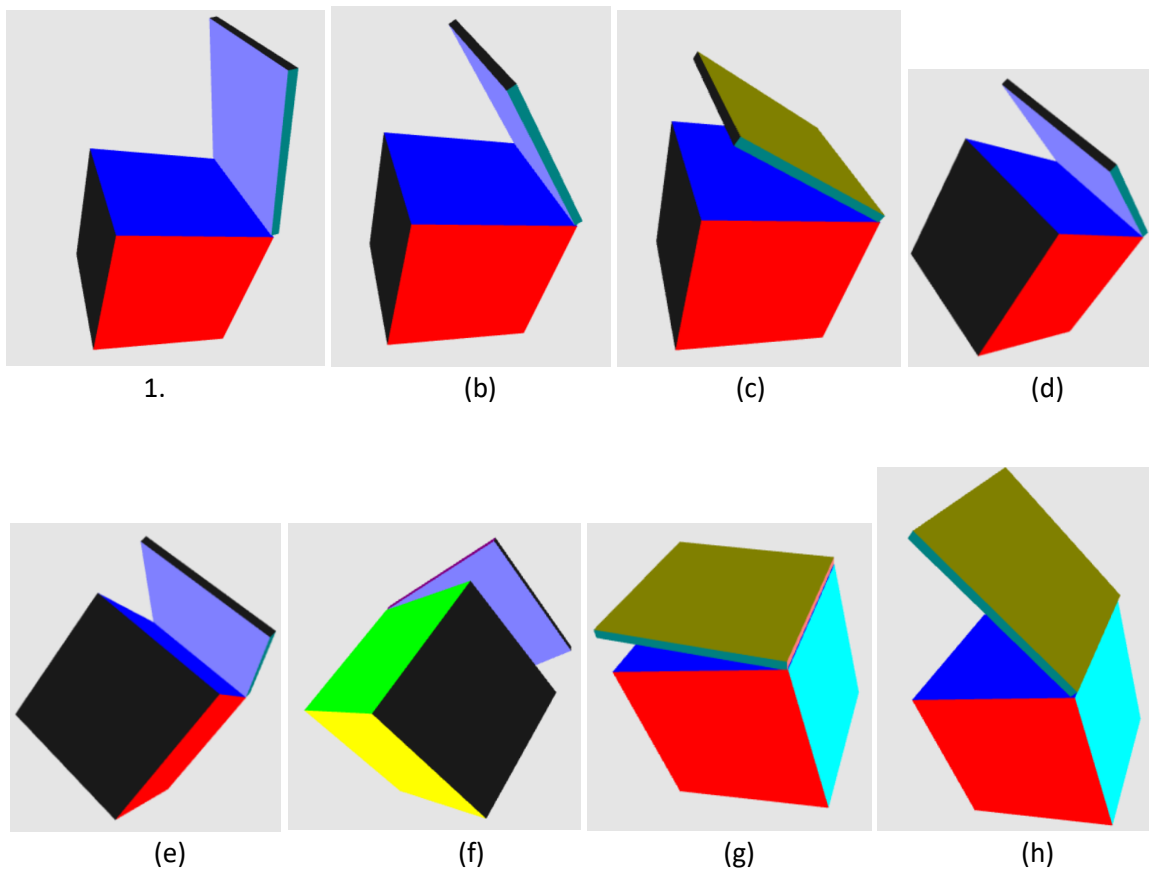


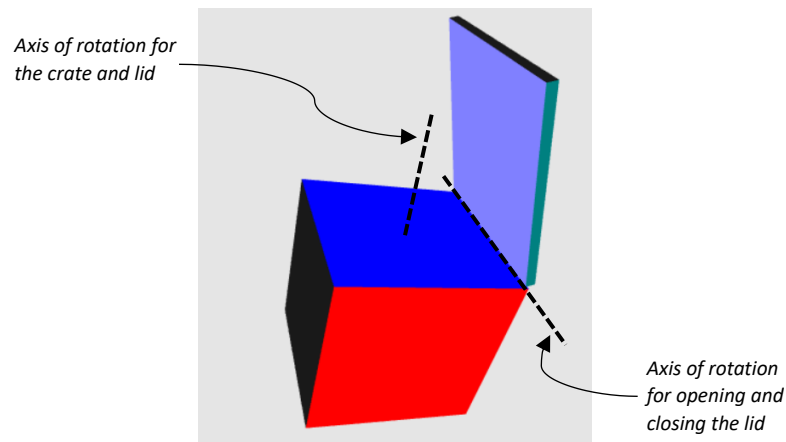
Figure: Different states of the crate and the lid. (a) – (c): Pressing down arrow key several times. (d) – (f): Pressing right arrow key several times. (g): Pressing left arrow key several times. (h) :Pressing up arrow key several times. [Video demo is available here.](#)

Requirement:

1. Initially the (crate + lid) will create 30-degree w.r.t X and Y axis respectively.
2. Your program must be modularized into several functions. There must be two initGL functions – which are `initGL_1()` and `initGL_2()`; and two draw functions which are `draw_crate()` and `draw_lid()` respectively.
3. Your program must have different functions for different transformation matrices, for perspective and for processing attribute buffers. The code skeleton contains the format of the function [see next section].
 - a. Transformation function: For example, the `rotate_Y(thetaY, loc)` function will send a rotation matrix along Y axis with `thetaY` degree to the `loc` location that contains the matrix in shader. Here `loc` is defined inside `initGL` function.
 - b. Perspective function: The function `perspective(aspect, fov, near, far, loc)` will take the necessary parameters (`aspect`, `fov`, `near` and `far`) and the location `loc` to be assigned.
 - c. Attribute buffer: For example, `passAttribData(data, att_buffer, loc)` will take CPU data, the buffer `att_buffer` which is created inside `initGL` and the location `loc` where the data needs to be passed.

Note:

1. Start working with the perspective first.
2. Carefully plan for the **composite transformation**. It is very crucial for this assignment.
3. Be careful on the axis of rotation for crate and lid. Note that, for rotating (crate + lid), the axis of rotation is actually the Y axis which goes through the center of the crate. However, for the opening and closing the lid, the axis of rotation is on of its edges. In that case, you might need to translate that edge to any of the principal axis (Z axis would be better), apply the rotation and undo the translation.
4. `gl.useProgram(prog)` is very important for switching from working between (crate + lid) and only lid.



Code skeleton:

[\[download the full version\]](#)

```
var vertexShaderSource_1 = ``;
var fragmentShaderSource_1 = ``;
var vertexShaderSource_2 = ``;
var fragmentShaderSource_2 = ``;
function passAttribData(data, att_buffer, loc) { }
function translate(tx, ty, tz, loc) { }
function scale(sx, sy, sz, loc) { }
function rotate_X(thetaX, loc) { }
function rotate_Y(thetaY, loc) { }
function rotate_Z(thetaZ, loc) { }
function perspective(aspect, fov, near, far, loc) { }
function model() { }
function draw_crate() {}
function draw_lid() {}
function createProgram(gl, vsSource, fsSource){}
function initGL_1(){}
function initGL_2() {}
function init() {}
init();
```

Deadline: 3rd March 2021 for both the sections. Google Forms will be provided for submission.

Evaluation: Coding + Viva