

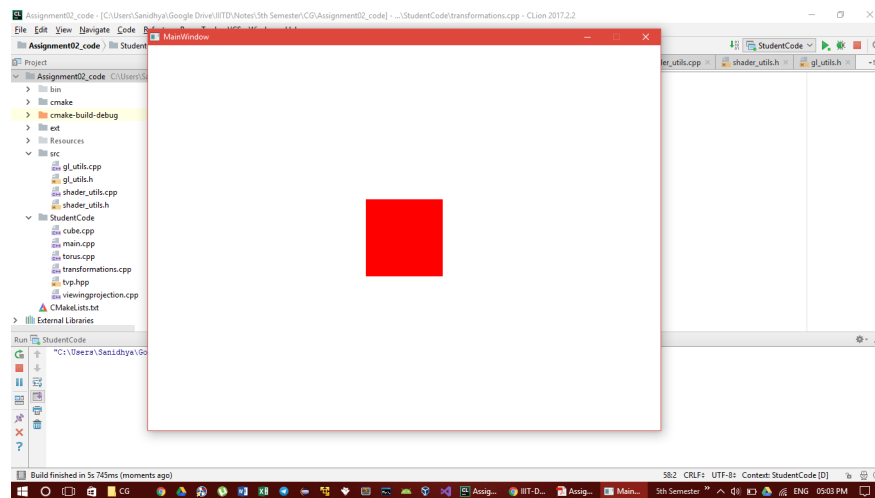
Assignment 2: Transforming, Viewing and Projection

SANIDHYA SINGAL*, Indraprastha Institute of Information Technology Delhi

ACM Reference format:

Sanidhya Singal. 2017. Assignment 2: Transforming, Viewing and Projection. *J. ACM* 1, 1, Article 1 (September 2017), 8 pages. <https://doi.org/0000001.0000001>

Initial Run:



*This is the corresponding author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2017 Association for Computing Machinery.

0004-5411/2017/9-ART1 \$15.00

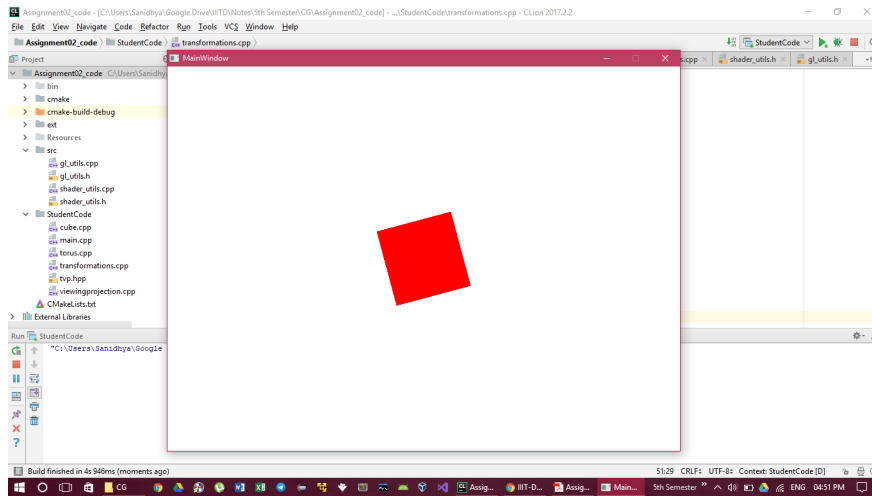
<https://doi.org/0000001.0000001>

1 TRANSFORMATION

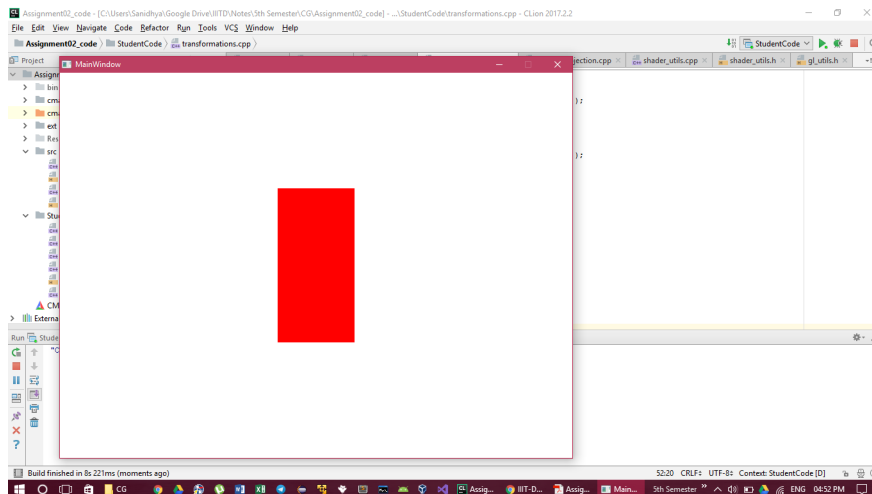
Ans 1

The screen-shots are shown below:

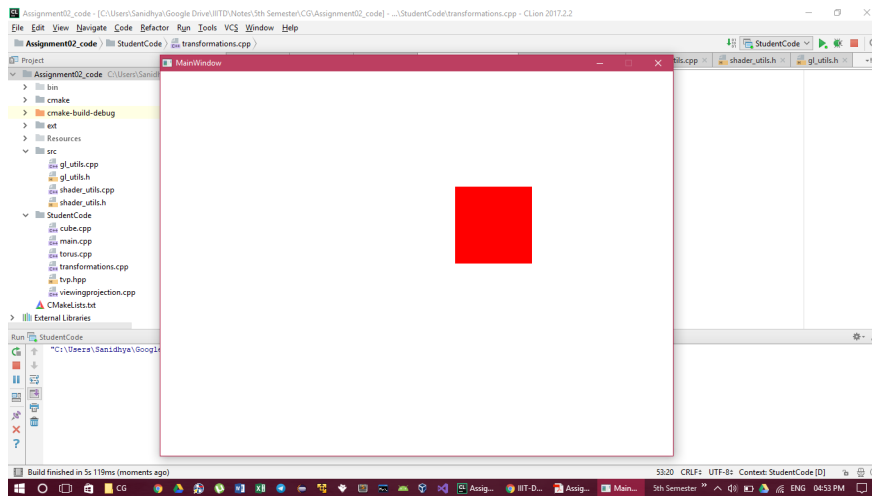
- (1) Rotate about z-axis by 15 degrees



- (2) Scale along y-axis by 2.0



(3) Translate by (20, 10)

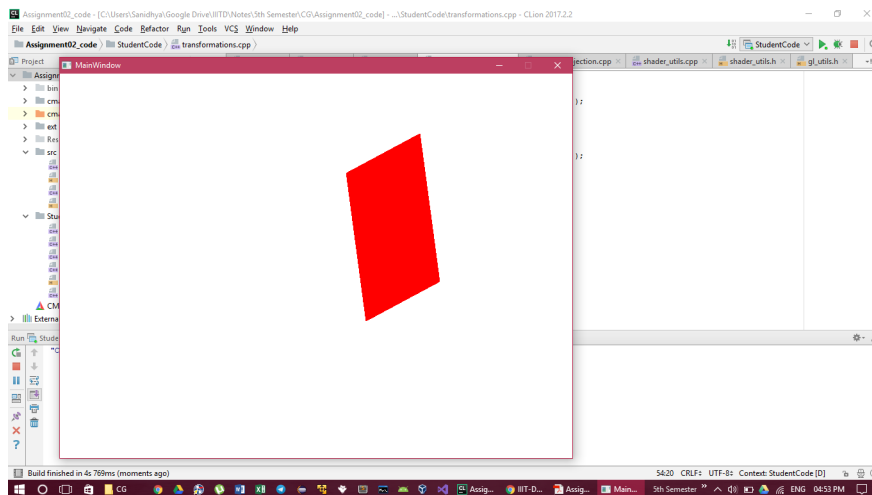


Ans 2

The screen-shots are shown below. The two outputs are not identical. The reason behind this is that matrix multiplication is NOT commutative. $AB \neq BA$

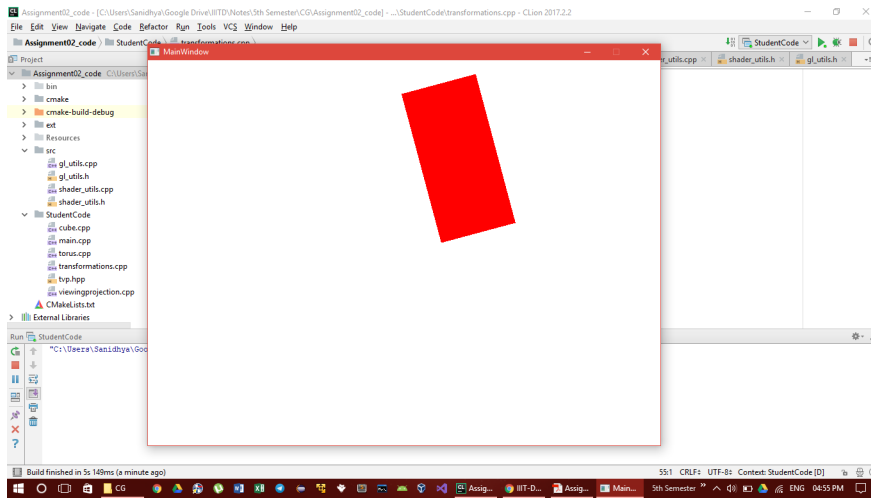
(1) $\langle a, b, c \rangle$

Here, we first rotate the object along z-axis by 15 degrees, then scale it along y-axis by 2.0 and finally, translate the result by (20, 10). Following is the result obtained:



(2) $\langle c, b, a \rangle$

Here, we first translate the object by (20, 10), then scale it along y-axis by 2.0 and finally, rotate the object along z-axis by 15 degrees. Following is the result obtained:



2 VIEWING

Ans 3

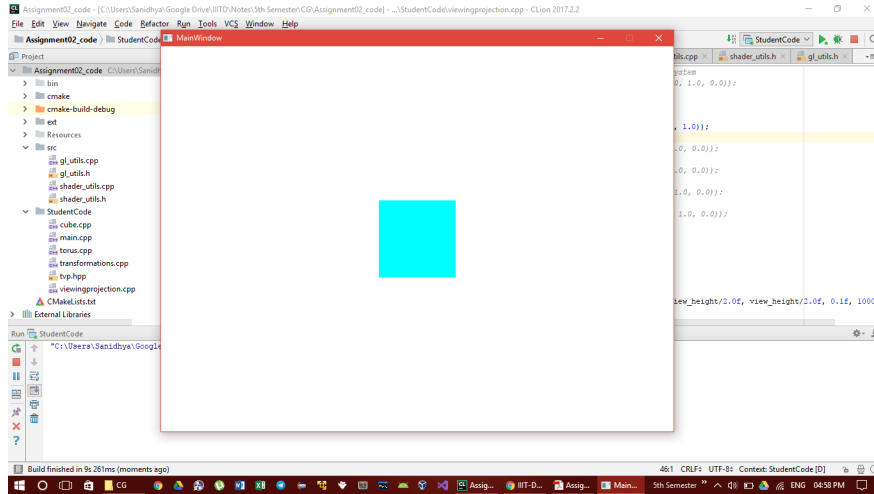
The `myLookAt()` and `glm::lookAt()` routines generate identical results.

3 PROJECTION

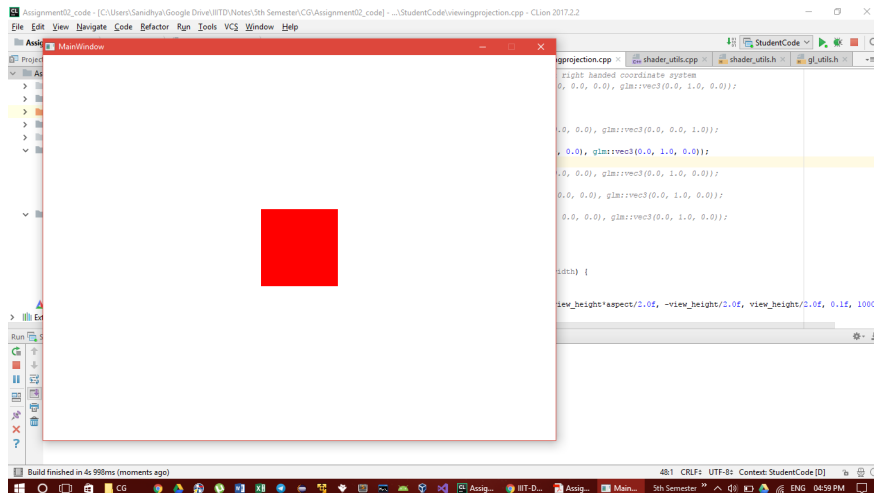
Orthographic Projection

We have a camera that is looking down the negative z-axis in a right handed coordinate system.

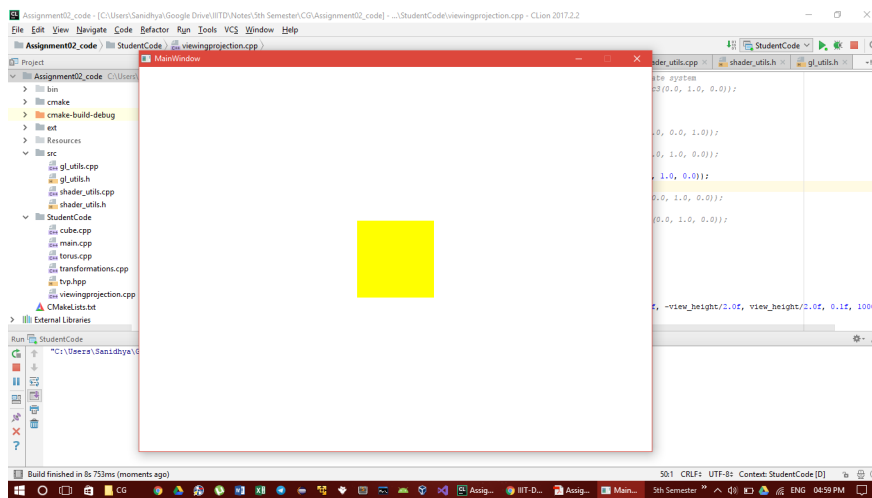
- (1) **Top:** The eye of the camera is at (0,100,0)



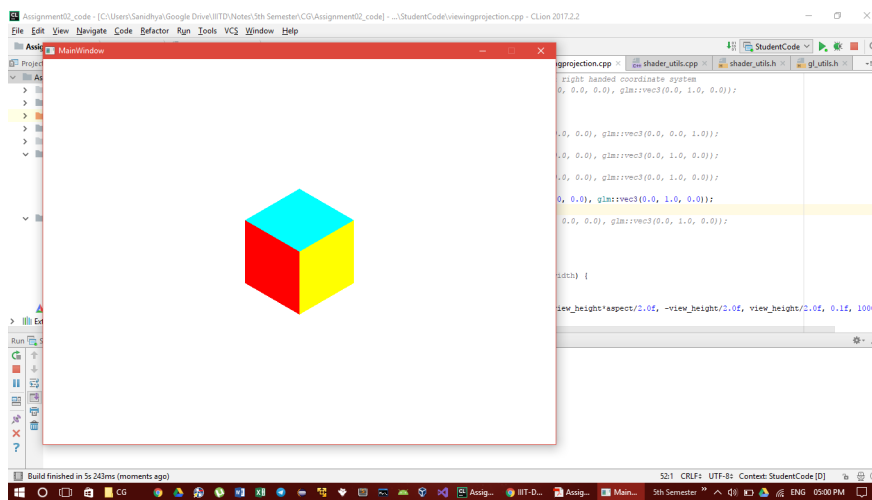
- (2) **Front:** The eye of the camera is at (0,0,100)



(3) **Side:** The eye of the camera is at (100,0,0)



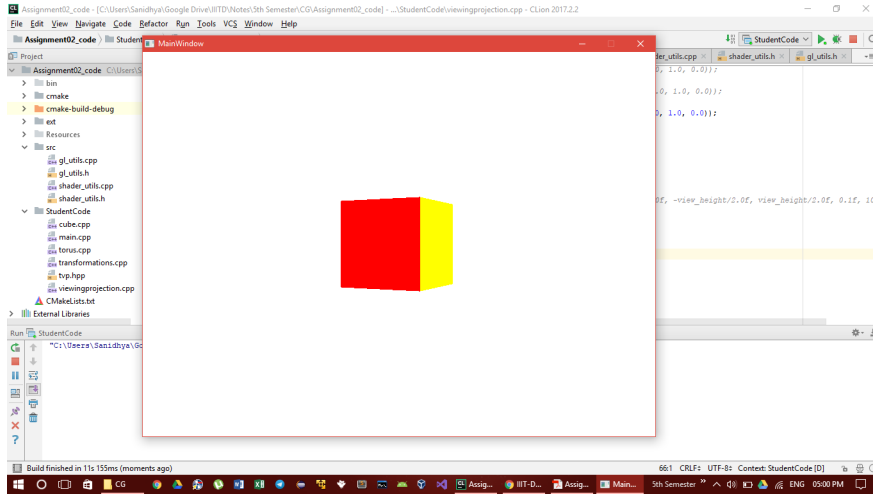
(4) **Isometric:** The eye of the camera is at (50,50,50)



Prespective Projection

The eye of the camera is at (50,0,100)

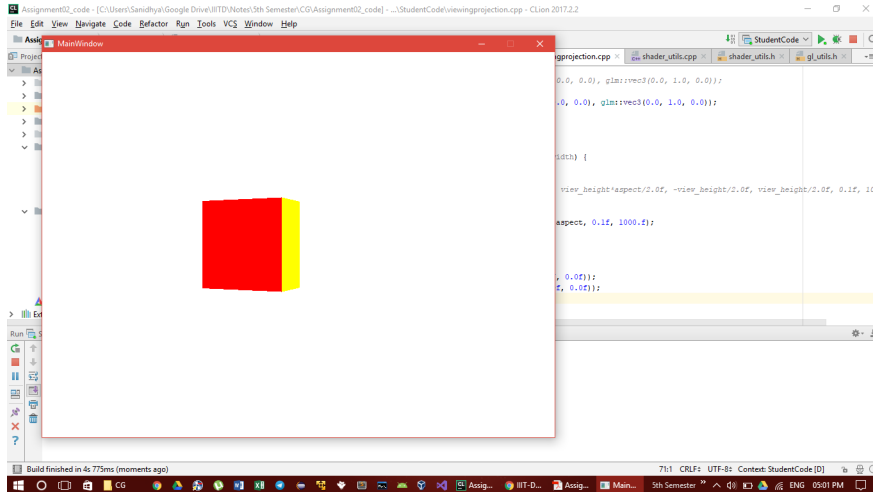
(1) One-point



(2) Two-point

Two-point projection is obtained from one-point projection as follows [1]:

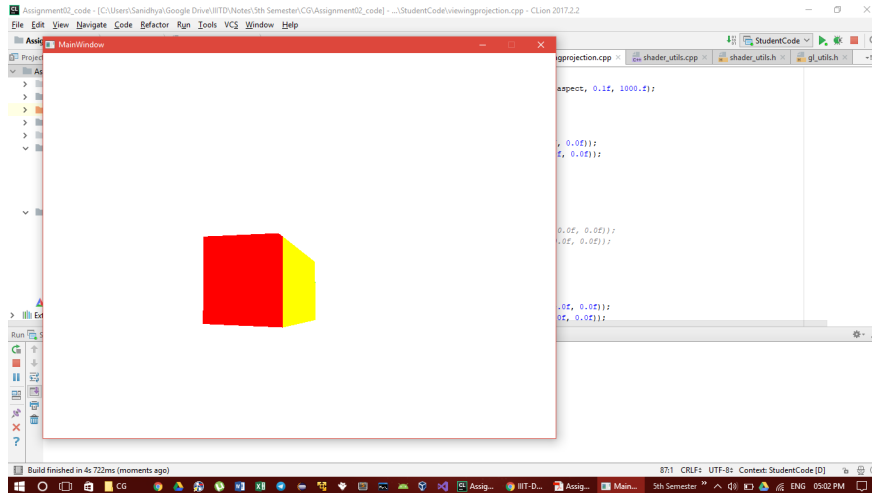
$$P_2 = R_y(-\theta) \cdot P_1 \cdot R_y(\theta)$$



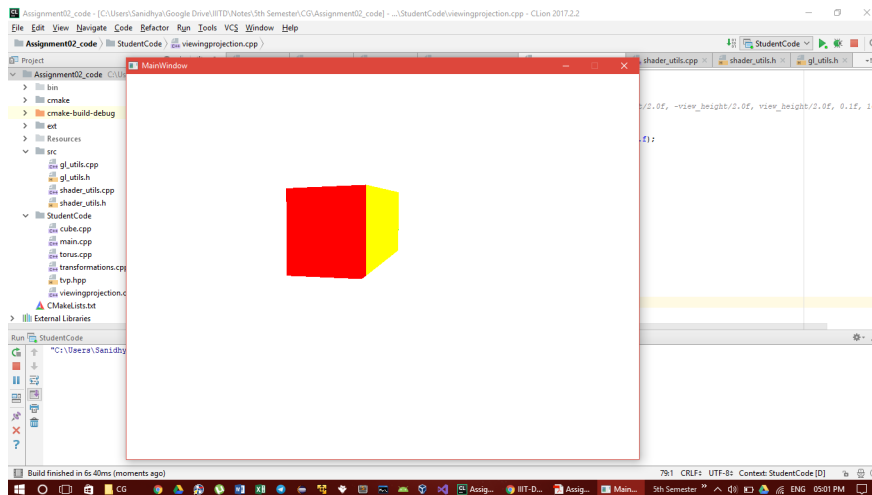
Three-point projection is obtained from two-point projection as follows [1]:

$$P_3 = R_x(-\phi) \cdot P_2 \cdot R_x(\phi)$$

(3) Three-point (Bird's Eye View)



(4) Three-point (Rat's Eye View)



REFERENCES

- [1] The Mathematics of Two- and Three- Point Perspective
<https://people.eecs.berkeley.edu/~barsky/perspective.html>