CS30800
Introduction to Computer Graphics
Lab 3 – Transformation

2025. 03. 18 / 2025. 03. 20

- Note: we will not answer questions directly related to the assignment
  - The goal is to find solutions based on your understandings for the course.

- In the lab session, we will provide explanations about
  - Purpose of the assignment
  - What you need to implement
  - Recap for the background knowledge

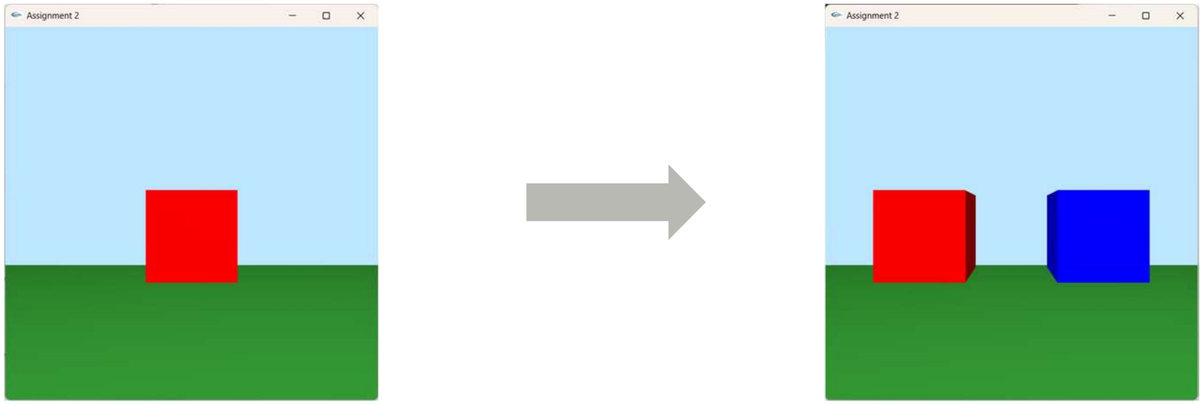
- Some slides are from lecture notes of this course

Contents

- HW2 goals and environment setup
- Recap: transformation and frames
- Overview: homework 2
- Q&A

HW2 Goals

- Draw two cubes



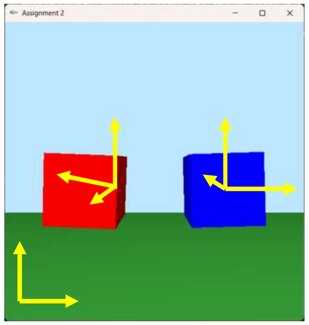
- Implement matrix operations for affine transformations

$$\begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & d \\ 0 & 1 & 0 & h \\ 0 & 0 & 1 & l \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a & b & c & 0 \\ e & f & g & 0 \\ i & j & k & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

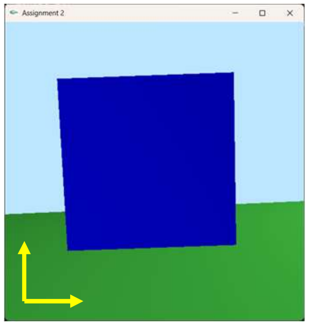
$$A = TL$$

HW2 Goals

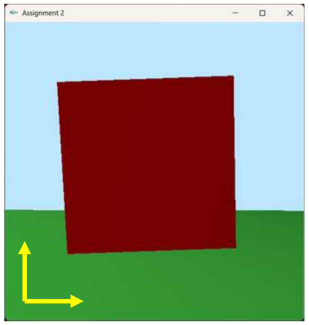
- Change viewpoint



Sky frame

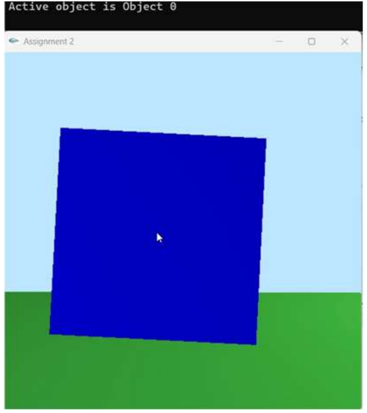


Red cube frame

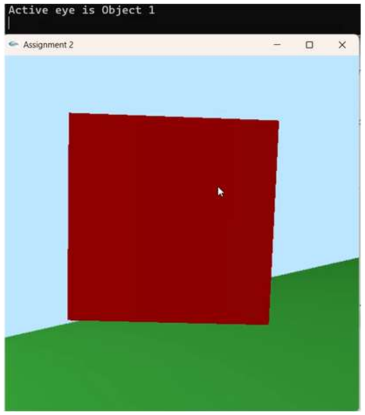


Blue cube frame

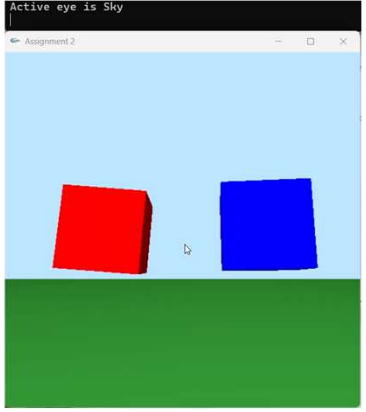
- Object manipulation (translation and rotation) in each frame.



```
Active object is Object 0
```



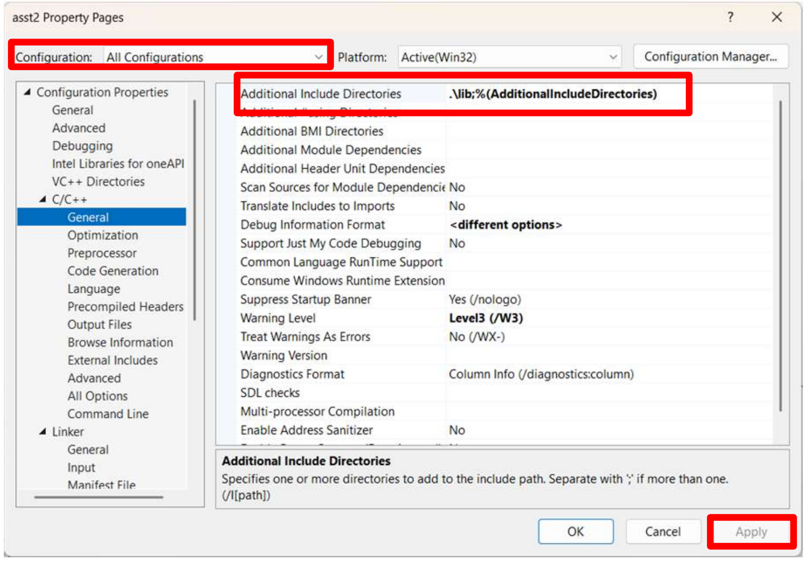
```
Active eye is Object 1
```



```
Active eye is Sky
```

Frequently Asked Questions – Setup

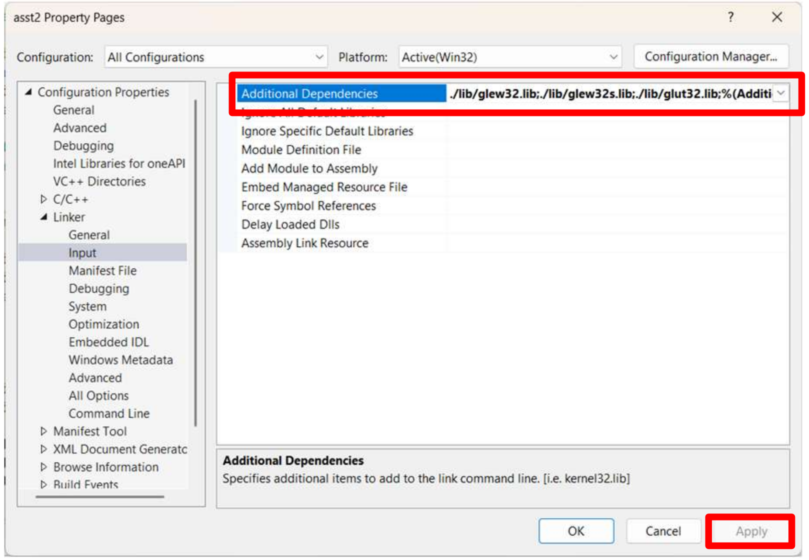
- Compilation failed on Windows VS2022
  - Add paths to the Properties → C/C++ → general → Additional Include Directories
    - .\lib;%(AdditionalIncludeDirectories)
  - Add paths to the Properties → Linker → Input → Additional Dependencies
    - ./lib/glew32.lib;./lib/glew32s.lib;./lib/glut32.lib;%(AdditionalDependencies)



Configuration: All Configurations
Platform: Active(Win32)

Additional Include Directories	**.\lib;%(AdditionalIncludeDirectories)**
Additional BMI Directories	
Additional Module Dependencies	
Additional Header Unit Dependencies	
Scan Sources for Module Dependencie	No
Translate Includes to Imports	No
Debug Information Format	**<different options>**
Support Just My Code Debugging	No
Common Language RunTime Support	
Consume Windows Runtime Extension	
Suppress Startup Banner	Yes (/nologo)
Warning Level	**Level3 (/W3)**
Treat Warnings As Errors	No (/WX-)
Warning Version	
Diagnostics Format	Column Info (/diagnostics:column)
SDL checks	
Multi-processor Compilation	
Enable Address Sanitizer	No

Additional Include Directories
Specifies one or more directories to add to the include path. Separate with ';' if more than one. (/I[path])



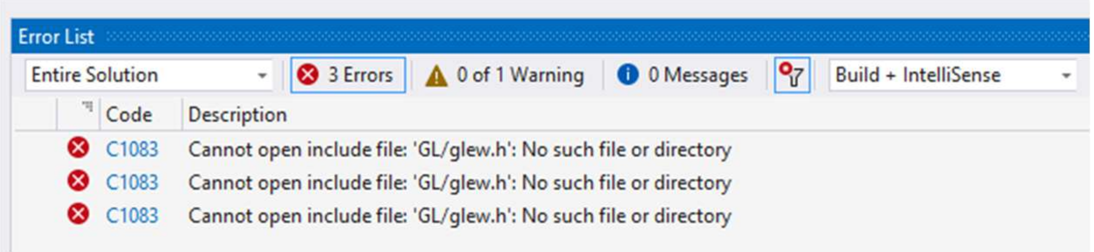
Configuration: All Configurations
Platform: Active(Win32)

Additional Dependencies	**./lib/glew32.lib;./lib/glew32s.lib;./lib/glut32.lib;%(Additi**
Ignore Specific Default Libraries	
Module Definition File	
Add Module to Assembly	
Embed Managed Resource File	
Force Symbol References	
Delay Loaded Dlls	
Assembly Link Resource	

Additional Dependencies
Specifies additional items to add to the link command line. [i.e. kernel32.lib]

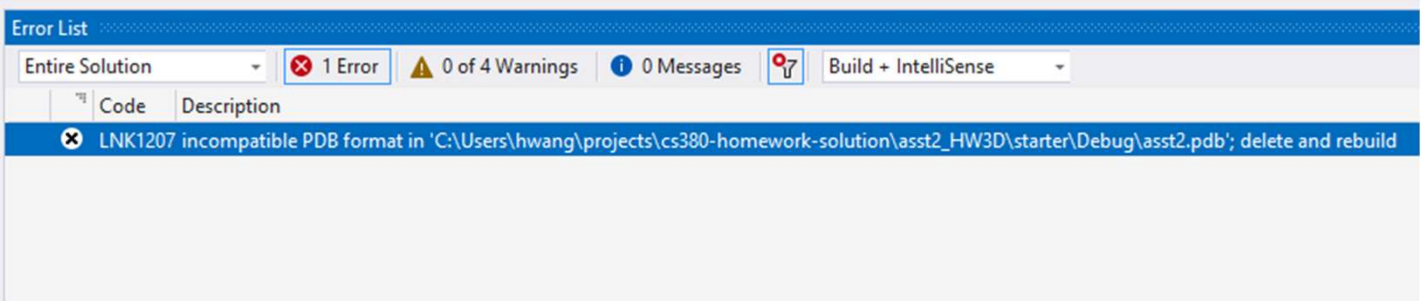
Frequently Asked Questions – Setup

- ‘GL/glew/h’: No such file or directory
  - Copy and paste from ‘lib’ folder from your previous assignment (HW1)



Code	Description
C1083	Cannot open include file: 'GL/glew.h': No such file or directory
C1083	Cannot open include file: 'GL/glew.h': No such file or directory
C1083	Cannot open include file: 'GL/glew.h': No such file or directory

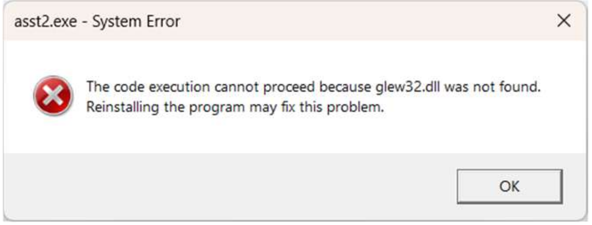
- Incompatible PDB format in ~~
  - delete *.pdb and rebuild project.



LNK1207 incompatible PDB format in 'C:\Users\hwang\projects\cs380-homework-solution\asst2_HW3D\starter\Debug\asst2.pdb'; delete and rebuild

Frequently Asked Questions – Setup

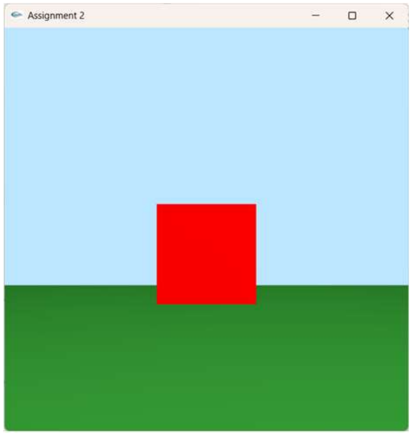
- *.dll not found
  - Copy and paste *.dll files from your previous assignment

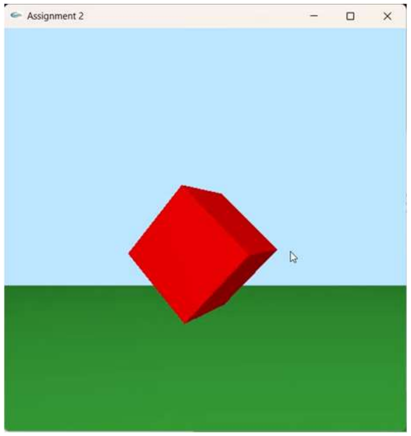


asst2.exe - System Error

The code execution cannot proceed because glew32.dll was not found. Reinstalling the program may fix this problem.

- If you solve the issues previously mentioned, you can start HW2!





Frequently Asked Questions – Setup

- Compilation failed on macOS
  - If you compile your assignment using `make` you can modify Makefile
  - You can use both clang++ and g++ compilers

- GLEW version mismatch
  - If you installed through `brew`, the version will be 2.2.0
    - Path for *.dylib: `/opt/homebrew/lib`
  - If you installed through `sourceforge`, the version will be 2.1.0 or others
    - Path for *.dylib: `/usr/local/lib`
  - Please set the GLEW lib path according to your preference.

Recap: Transformations and Frames

Affine transformation

Full affine transformation Translation Rotation

$$\begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & d \\ 0 & 1 & 0 & h \\ 0 & 0 & 1 & l \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a & b & c & 0 \\ e & f & g & 0 \\ i & j & k & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

$$\begin{bmatrix} l & t \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} i & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} l & 0 \\ 0 & 1 \end{bmatrix}$$

$$A = TL$$

Transformation Respect

- We are transforming a point $\tilde{p}$ in a frame $\vec{\mathbf{f}}^t$

$$\tilde{p} = \vec{\mathbf{f}}^t \mathbf{c}$$

- With a matrix

$$S = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

the stretches by factor of two in first axis of $\vec{\mathbf{f}}^t$

- Performing a transform: $\vec{\mathbf{f}}^t \mathbf{c} \Rightarrow \vec{\mathbf{f}}^t S \mathbf{c}$

- Suppose another frame: $\vec{\mathbf{a}}^t = \vec{\mathbf{f}}^t A$

Transformation Respect

- We could express the point with a new coordinate vector

$$\tilde{p} = \vec{\mathbf{f}}^t \mathbf{c} = \vec{\mathbf{a}}^t \mathbf{d} \qquad \vec{\mathbf{a}}^t = \vec{\mathbf{f}}^t A$$

$$\vec{\mathbf{f}}^t \mathbf{c} = \vec{\mathbf{f}}^t A\mathbf{d} \qquad \vec{\mathbf{f}}^t = \vec{\mathbf{a}}^t A^{-1}$$

$$\mathbf{d} = A^{-1}\mathbf{c}$$

- Now S transforms the point $\tilde{p}$ with respect to $\vec{\mathbf{a}}^t$

$$\vec{\mathbf{a}}^t \mathbf{d} \Rightarrow \vec{\mathbf{a}}^t S\mathbf{d}$$

Transformation Respect

- Point is transformed with respect to the the frame that appears immediately to the left of the transformation matrix in the expression.
- We read

$$\vec{\mathbf{f}}^{t} \Rightarrow \vec{\mathbf{f}}^{t} S$$

$\vec{\mathbf{f}}^{t}$ is transformed by S with respect to $\vec{\mathbf{f}}^{t}$

- We read

$$\vec{\mathbf{f}}^{t} = \vec{\mathbf{a}}^{t} A^{-1} \Rightarrow \vec{\mathbf{a}}^{t} S A^{-1}$$

$\vec{\mathbf{f}}^{t}$ is transformed by S with respect to $\vec{\mathbf{a}}^{t}$

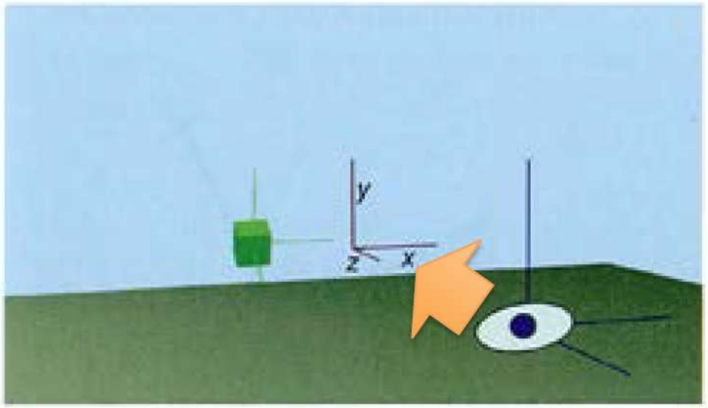
Frames

- World frame $\boxed{\vec{w}^t}$
  - an absolute frame in 3D space
  - All other frames are represented by this frame
- Object frame $\boxed{\vec{o}^t}$
  - All objects should have own frame

$$\vec{o}^t = \vec{w}^t O$$

- Eye frame $\boxed{\vec{e}^t}$

$$\vec{e}^t = \vec{w}^t E$$



(a) The frames

Eye Coordinate

- we explicitly store the matrix E

$$\tilde{p} = \vec{\mathbf{o}}^t \mathbf{c} = \vec{\mathbf{w}}^t O\mathbf{c} = \vec{\mathbf{e}}^t E^{-1} O\mathbf{c}$$

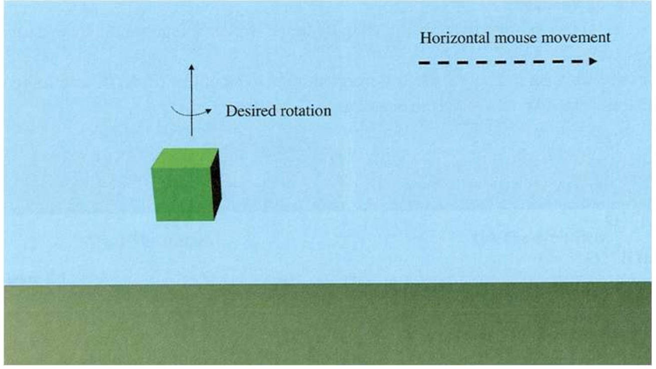
$\vec{\mathbf{o}}^t = \vec{\mathbf{w}}^t O,$

$\vec{\mathbf{e}}^t = \vec{\mathbf{w}}^t E$

  - Object coordinates: $\mathbf{c}$
  - World coordinates: $O\mathbf{c}$
  - Eye coordinates: $E^{-1}O\mathbf{c}$
  - Calculating the eye coordinates of every vertexes:

$$\begin{bmatrix} x_e \\ y_e \\ z_e \\ 1 \end{bmatrix} = E^{-1}O \begin{bmatrix} x_o \\ y_o \\ z_o \\ 1 \end{bmatrix}$$

Moving an Object



- We want to manipulate the object:
  - Translate the object along the mouse movement
  - Rotate the object w.r.t. y-axis of the eye frame but centered to the object
- Which frame should be selected to make the purpose?
  - Please refer to lecture slides!

Overview: Homework 2

Task 1 – Draw Two Cubes

- A frame need to be defined to each object.

- A frame of a red cube

$$\vec{o}^{\,t}_{R} = \vec{w}^{\,t}\, O_R$$

- A frame of a blue cube

$$\vec{o}^{\,t}_{B} = \vec{w}^{\,t}\, O_B$$

- You can simply copy and paste vertex coordinates in red cube.
  - Different frames allow the cubes to be positioned differently.

Task 2 – Matrix Operations

- linFact
  - Rotation matrix: 4 x 4
- transFact
  - Translation matrix: 4 x 4

- M = transFact(M) * linFact(M);

- By using these two function, you can obtain the full affine transformation matrix

Task 3 – Change Viewpoint

- The window visualizes the projection of 3D models onto the XY-plane of the eye frame.

- Initially, the rendered image is generated in the frame of the sky-view

- In this assignment, you need to adjust the eye-view based on user input by modifying the eye-view matrix.
(sky, cube1 and cube2)

Task 4 – Manipulation Mode

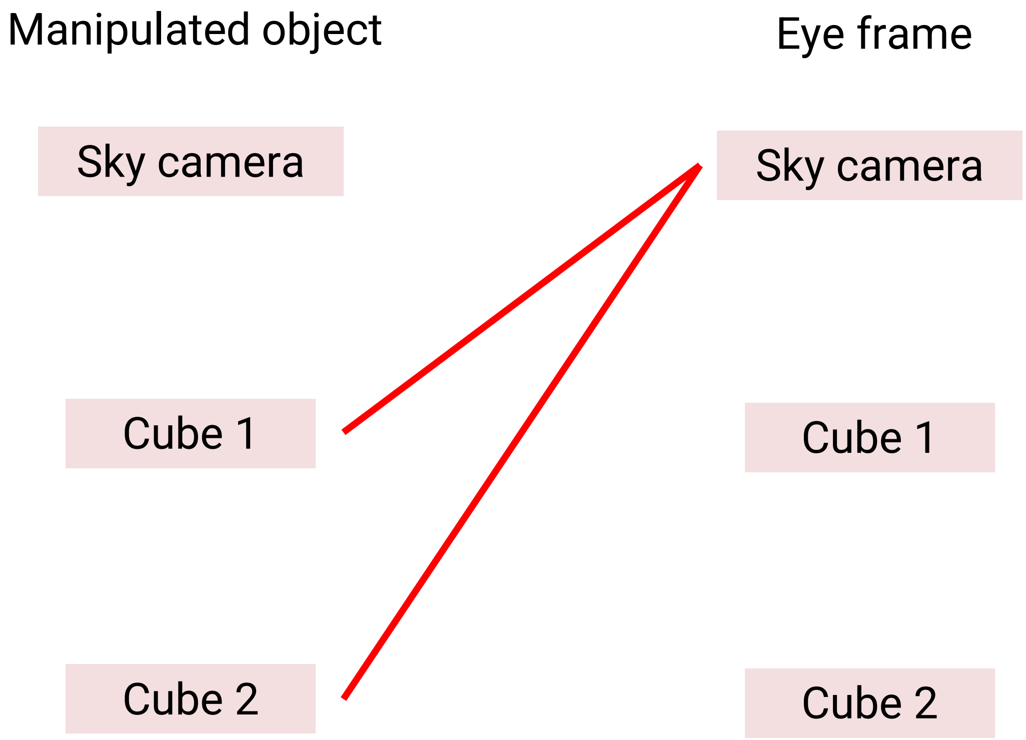
- If 'o' key is pressed, the object that we can manipulate should be changed (sky → cube1 → cube2)

- You should choose different frames carefully depending on object and eye mode (there are 8 combinations)

- In order to complete task 4, you should utilize a transformation w.r.t. a frame.

Task 4 – Manipulation Mode

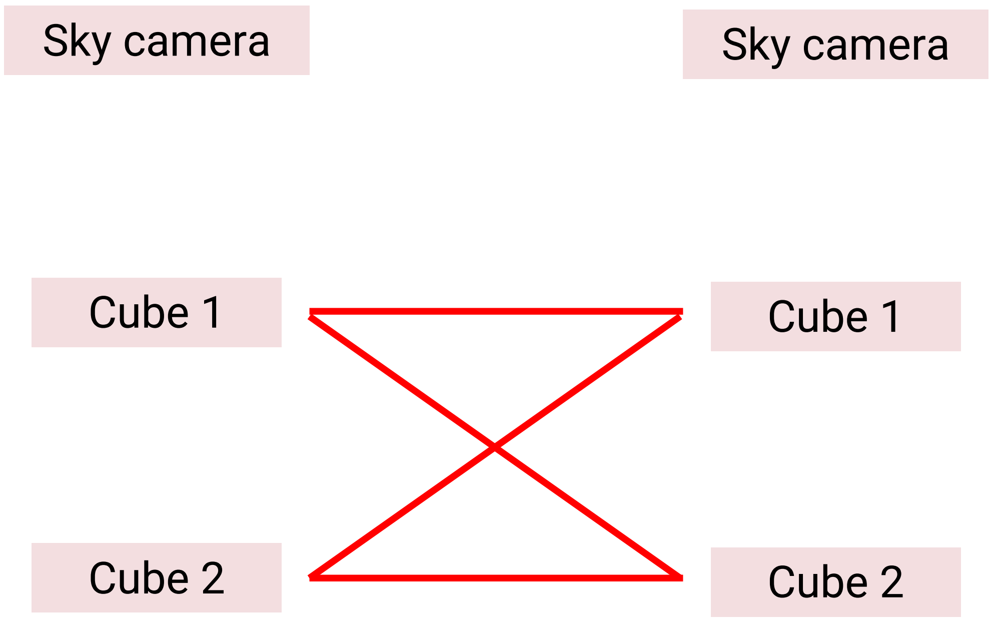
$\vec{a}^{t}$ should be the cube-sky frame



Task 4 – Manipulation Mode

$\vec{a}^{t}$ should be the cube-cube frame

Manipulated object



Sky camera

Sky camera

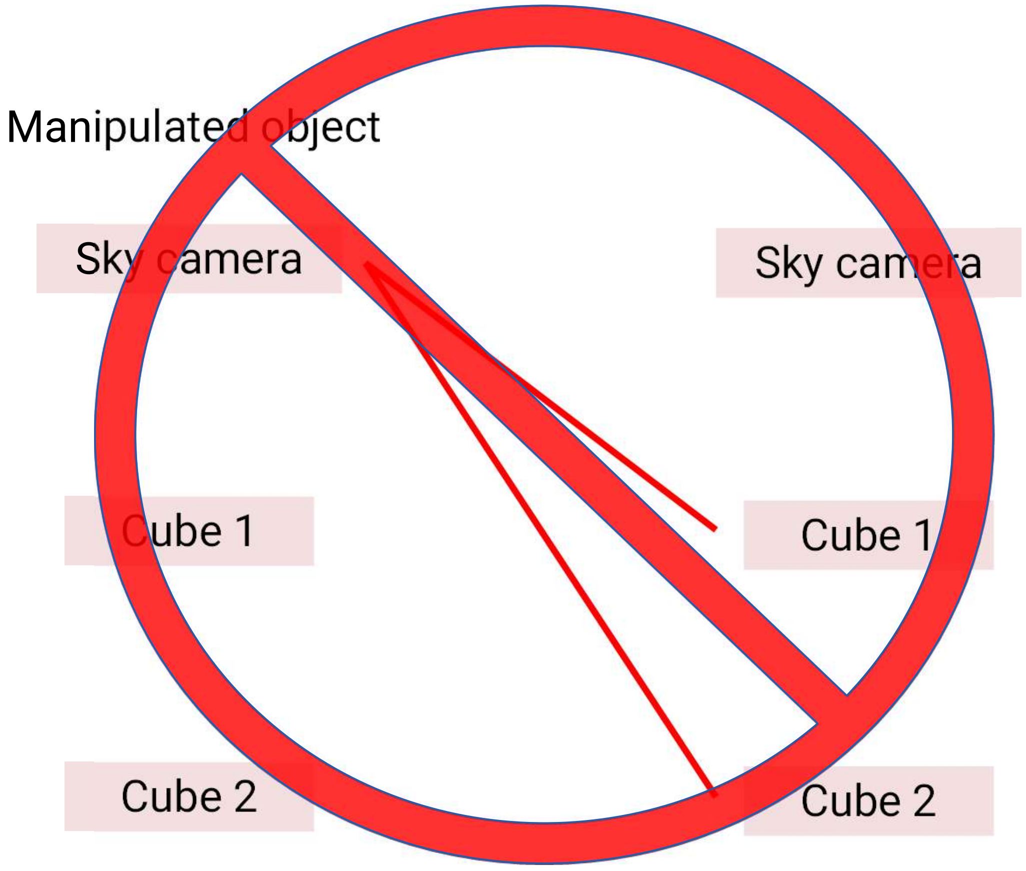
Cube 1

Cube 1

Cube 2

Cube 2

Task 4 – Manipulation Mode



Task 4 – How to choose frame for manipulations?

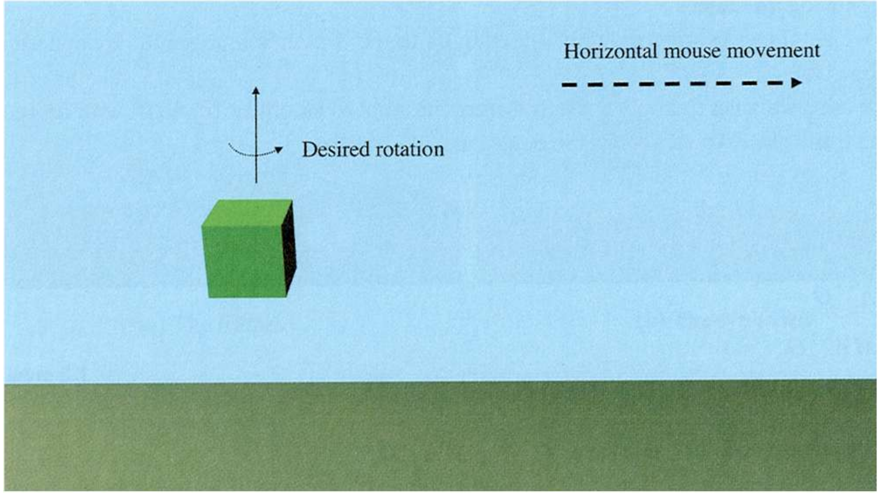
- Translation: direction
- Rotation: position (origin) + direction (axis)

- Recalling the Affine transformation: $A = TR$
- The object's Affine transformation: $O = (O)_T \, (O)_R$
- The eye's Affine transformation: $E = (E)_T \, (E)_R$

- How to construct auxiliary frame for the intuitive object manipulation?

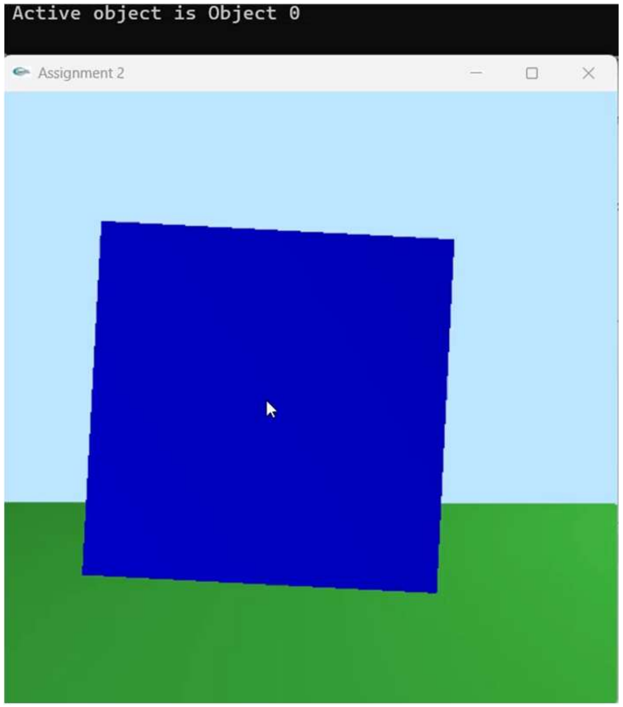
Task 4 – Move Objects

- You need to focus…
  - How to make the objects rotate and translate properly?
  - What matrices need to be updated when the viewpoint or manipulated object changes?
  - How can restrict adjustments that prohibited(e.g. manipulating sky-cam)?

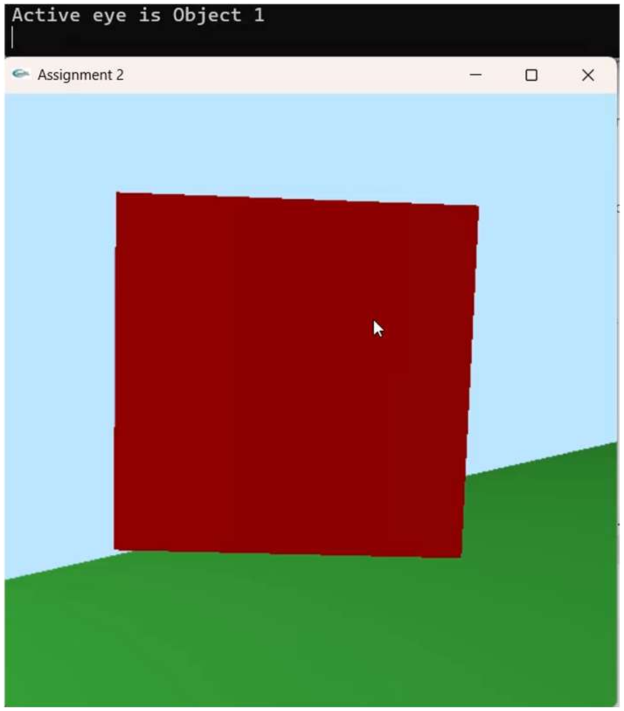


Please refer to ‘slide07-hello3D’ before starting programming!

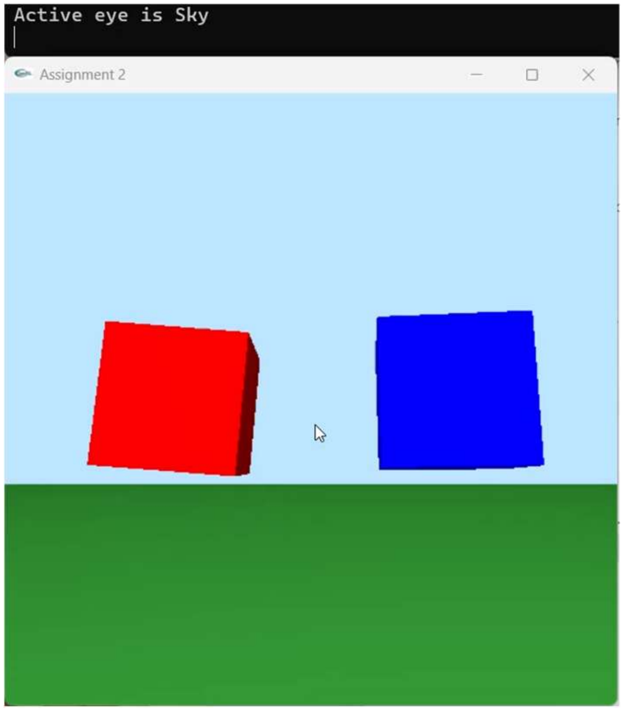
Desired behaviors of the homework 2



```
Active object is Object 0
```



```
Active eye is Object 1
```



```
Active eye is Sky
```