

Report:

Files:

- src
- —main.cpp
- —Helps.cpp
- —Helps.h

Only Eigen is used Here. glm is used at all.

Clear all contents in build directory if you do not use Xcode, make and run bin file.

—object basic control:

- '1' add a cube
- '2' add a bunny
- '3' add a bumpy cube
- 'r' cancel selection

('f' delete the selected item)

—render control:

- '4' wireframe
- '5' flat shading
- '6' phong shading
- '7' default shading

—object movement & size control:

- 'a' translation left (x axis)
- 'd' translation right
- 'w' translation up (y axis)
- 's' translation down
- 'q' translation backward (z axis)
- 'e' translation forward
- 'o' rotate according to y axis clockwise
- 'p' rotate according to y axis anti-clockwise
- 't' rotate according to x axis clockwise
- 'y' rotate according to x axis clockwise
- 'u' rotate according to z axis clockwise
- 'i' rotate according to z axis clockwise

'k' enlarge instance

'l' shrink instance

—camera control:

'z' translation left (x axis)	/	r enlarge
'x' translation right	/	r shrink
'c' translation up (y axis)	/	phi enlarge
'v' translation down	/	phi shrink
'b' translation backward (z axis)	/	theta enlarge
'n' translation forward	/	theta shrink

—perspective/orthogonal:

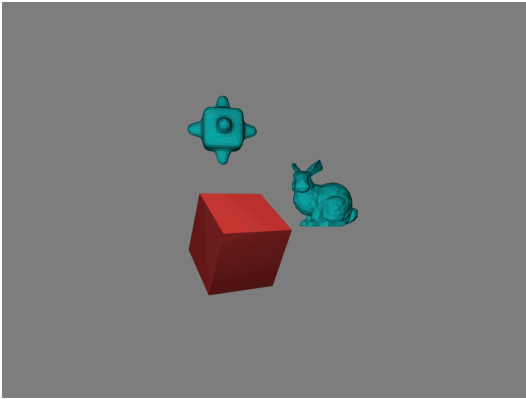
'0' for switching between two systems.

'9' activate track ball camera control

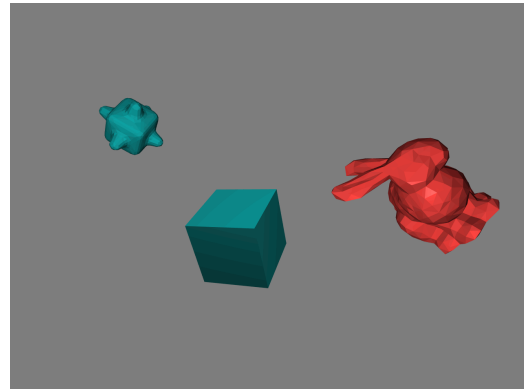
Visual Effect Demo:

Simple Display:

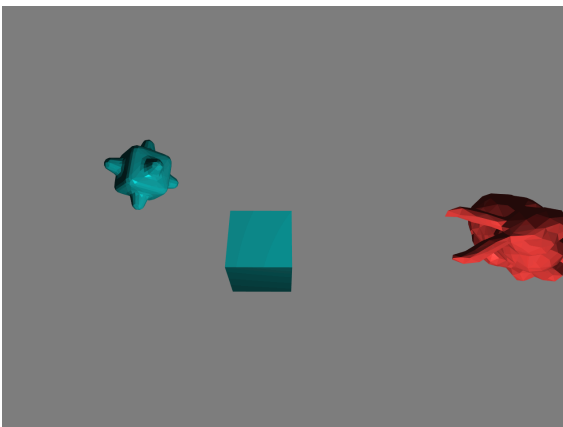
I create them on different spot
on purpose for better visualisation result.



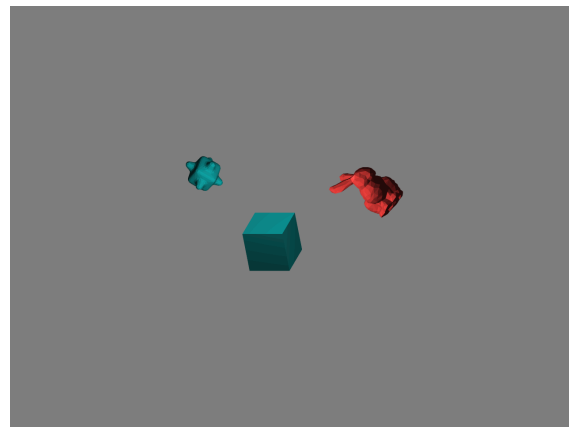
After some rotation and translation and
scale change, items can only be modified
when selected.



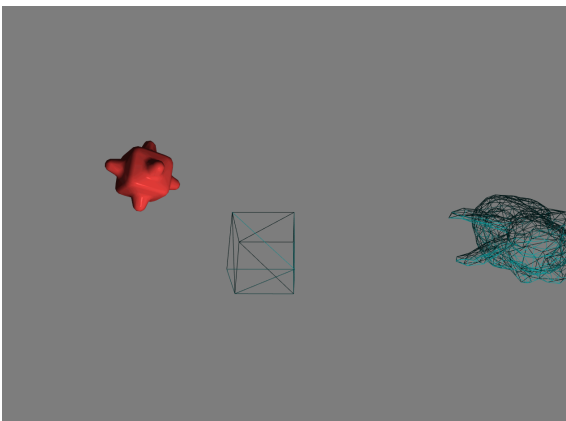
I can also translate my camera in x,y,z directions.
The camera always points towards the (0,0,0)



Switch to Orthogonal from perspective.



Demonstrate different shading settings;
Original, Wireframe, Phong shading.



Demonstrate different shading settings;
Flat shading, phong shading.

