In this assignment you will be implementing the Look-At, Turn-To, and Mouse-Look algorithms from lecture. You will demonstrate the functionality of your implementations in your interactive 3D graphics application.

Setup

This lab will be building off of your previous labs

- Declare three new 4x4 matrices before your main loop
    - These matrices will be used as global/worldspace matrices for three objects in the scene
    - Draw these matrices using the debug renderer
    - Draw each as three line-segments: X-Axis (Red), Y-Axis (Green), Z-Axis (Blue)

Windows Messages for Basic Input

Add cases to the switch statement in your WndProc function to handle mouse and keyboard input events. These events may have key-repeat delay built in, so you cannot depend on them to tell you that a key is down every cycle that it's held.

- Implement a table of boolean values representing key/button states.
- Set the bool for a key/button to true on a 'DOWN' message, false on an 'UP'
- See std::bitset for a compact way of representing a table of bools

Track mouse movement

- Track a previous position for the mouse
- Use the delta from previous to current position to track relative movement
- See the official documentation for the WM_MOUSEMOVE

Messages

- WM_KEYDOWN, WM_KEYUP WM_LBUTTONDOWN, WM_LBUTTONUP
- WM_RBUTTONDOWN, WM_RBUTTONUP WM_MOUSEMOVE

Keys (wParam)

- VK_LEFT, VK_RIGHT, VK_UP, VK_DOWN (arrows)
- 'W' 'A' 'S' 'D', etc. (letter keys)

Test Your Input

In the main update loop, use input to manipulate one of the matrices to serve as a moving target

- Up-Arrow Key : Translates the matrix along its local +Z
- Down-Arrow Key : Translates the matrix along its local -Z
- Left-Arrow Key : Rotates the matrix around its local Y to turn left
- Right-Arrow Key : Rotates the matrix around its local Y to turn right

LOOK-AT - 25%

- Do not use the DirectXMath Look-At function
- Make one matrix look-at the moving target
- Input parameters:
  - Viewer position (V), Target position (T), Local Up (U)
- Output result:
  - A new 4x4 matrix positioned at V and looking at T
  - Make your target be in a lower elevation.

TURN-TO - 25%

- Make one matrix turn-to the moving target
- Input parameters:
  - Viewer 4x4 matrix (M), Target position (T), Speed scalar (s)
- Output result:
  - A new 4x4 matrix to replace M, rotated partially towards T
  - Make your target be in a lower elevation.

MOUSE-LOOK - 25%

- Implement Mouse-Look for the camera
- Input parameters:
  - Viewer 4x4 matrix (M), Delta scalars (dX, dY)
- Output result:
  - A new 4x4 matrix
- Rotate the matrix M about its X-Axis by dY
- Rotate the matrix M about its Y-Axis by dX
- Cancel all rolling on the Z-Axis

TESTING - 20%

- Support translating the camera on X and Z axes by keypress.
  - 'W' Key : Translates the camera along its local +Z
  - 'S' Key : Translates the camera along its local -Z
  - 'A' Key : Translates the camera along its local -X
  - 'D' Key : Translates the camera along its local +X

Use Update function - 5%

- Be sure to do all of this logic in the update function, not the draw_view