Programming Assignment 2

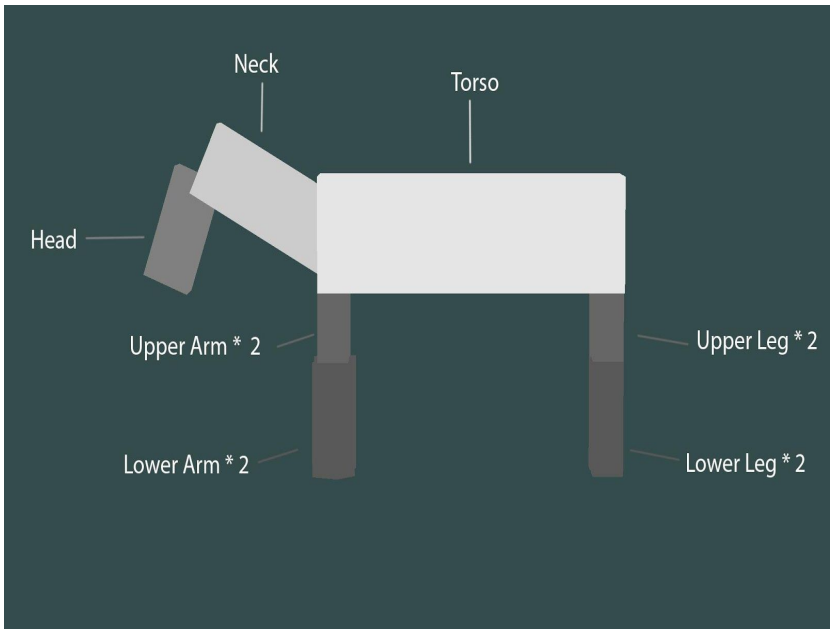
Announcement:	**Session # 5**
Submission Deadline:	**Session # 8**

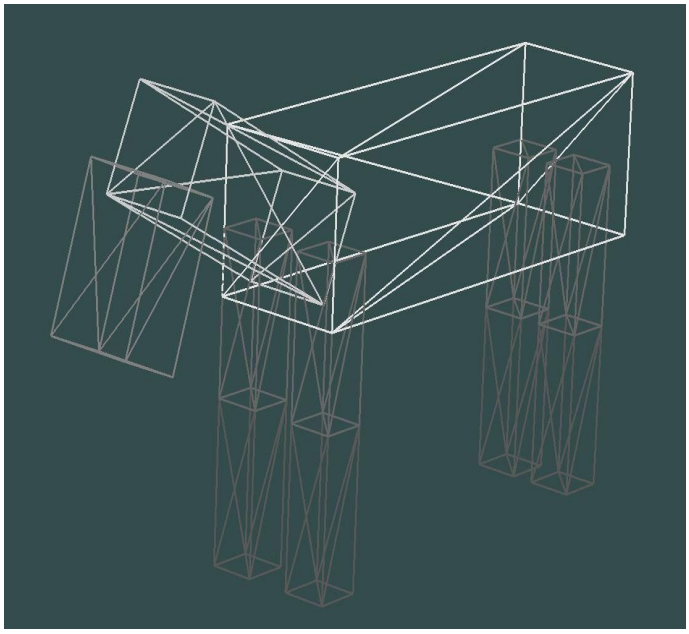
Description

This OpenGL programming assignment will build upon the horse modelling program you have developed as part of Assignment 1. If you have not completed Assignment 1, then you may do it as part of this assignment and continue with completing Assignment 2 in which case we will give you 3 of the 7 marks allotted to the first assignment. More specifically, you will learn about lighting, shading, texture mapping, shadow generation, and joint movement.

Please recall that your horse **mesh** is composed of head, neck, torso and legs. These are connected to each other through joints (axis) about which they can rotate relative to each other. Joints are named as follows:

1. head_to_neck,
2. neck_to_torso,
3. torso_to_front_upper_right_leg,
4. front right_knee,
5. torso_to_hind_upper_right_leg,
6. hind_right_knee,
7. torso_to_front_upper_left_leg,
8. front_left_knee,
9. torso_to_hind_upper_left_leg,
10. hind_left_knee





Implementation Specifications

- Extend your OpenGL application from Assignment 1 with the following functionality and features:
  - Illuminate the scene by adding a point light source (white) 20 units above the horse using the phong model.
  - Render the scene with grass texture on the ground mesh and horse-skin texture on the horse (Key X).
  - Render the scene with shadows using two pass shadow algorithm (Key B)
  - Rotate joint 0 by 5 degrees (Key_0 clockwise and the corresponding Shift + Key_0 for counterclockwise). Similarly for other numbered joints, that is Key_1 for joint 1, Key 2 for joint 2, etc.

- Please note the above are extensions to your program and previous key presses should work as before but with extended rendering as required in this assignment.
  - Pressing the spacebar should reposition the horse at a random location on the grid.
  - The user can incrementally size up the horse by pressing ‘U’ for scale-up and ‘J’ for scale-down. Each key press should result in a small size change.
  - The user can control the horse position and orientation using keyboard input i.e. A → move left 1 grid unit, D → move right 1 grid unit, W → move up 1 grid unit, S → move down 1 grid unit, a → rotate left 5 degrees about Y axis, d → rotate right 5 degrees about Y axis, w → rotate upwards 5 degrees raising the front legs, s → rotate downwards 5 degrees raising the hind legs.
  - The world orientation is changed by using keyboard input i.e. left arrow → R_x, right arrow → R_{-x}, up arrow → R_y, down arrow → R_{-y}. Pressing the “Home” button should reset to the initial world position and orientation.

- The user can change the rendering mode i.e. points, lines, triangles based on keyboard input i.e. key 'P' for points, key 'L' for lines, key 'T' for triangles.
- The user can pan and tilt the camera as follows:
  - while right button is pressed → use mouse movement in x direction to pan.
  - while middle button is pressed → use mouse movement in y direction to tilt. If no middle mouse, should be able to press both right and left button in order to tilt.
- The user can zoom in and out of the scene - while left button is pressed → use mouse movement to move into/out of the scene.
- Window resize handling: The application should handle window resize events and correctly adjust the aspect ratio accordingly. This means that the meshes should not be distorted in any way.

- The application should use OpenGL 3.0 and onwards, and include brief comments explaining each step.

Submission (electronic submission through Moodle only)

Please create a zip file containing your C/C++ code, vertex shader, fragment shader, a readme text file (.txt). In the readme file document the features and functionality of the application, and anything else you want the grader to know i.e. control keys, keyboard/mouse shortcuts, etc.

Additional Information

- A video demonstrating the functionality is posted on YouTube:
  https://youtu.be/sG0rnW08S5M

Extra Credit (20% points)

(1) Make the horse complete a run or walk cycle (Key R).

Evaluation Procedure

You MUST demonstrate your solution program to the lab instructor during lab hours. You must run your submitted code, demonstrate its full functionality and answer questions about the OpenGL programming aspects of your solution. Major marking is done on the spot during demonstration. Your code will be further checked for structure, non-plagiarism, etc. However, ONLY demonstrated submissions will receive marks. Other submissions will not be marked.