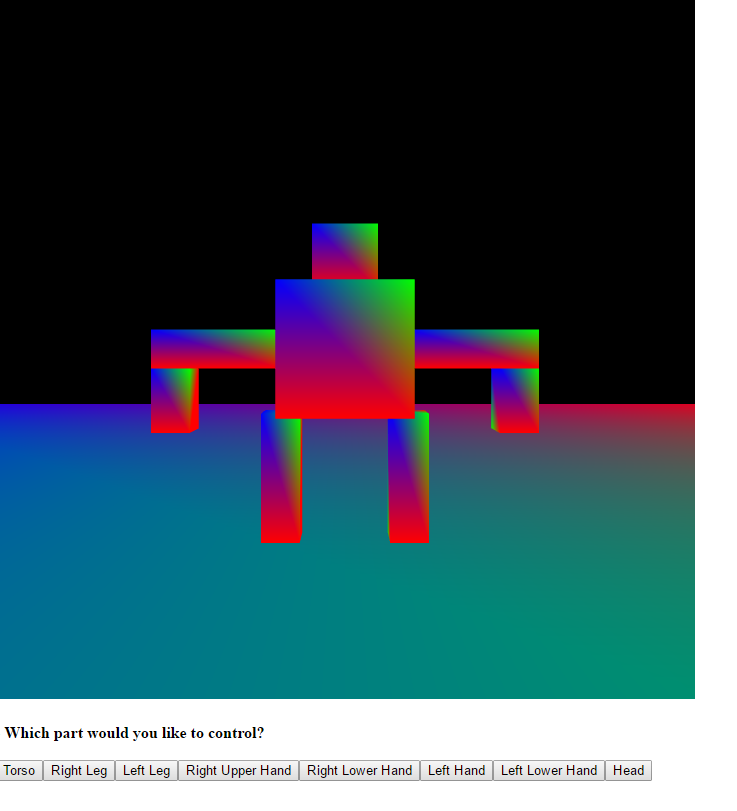
Readme of RT Rendering Lab2:

1. How to use the program:



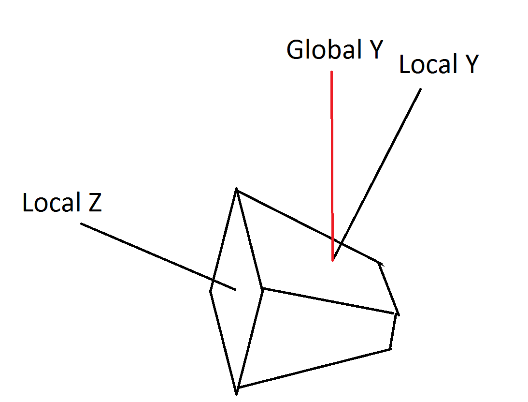
a. First click each button on the bottom of webpage to choose which part you would like to move.

b. To move a part, you can press:  
 F/ B: forward, backward  
 L/ R: left, right  
 Or conventional WSAD keys  
 Space/ C: upward, downward

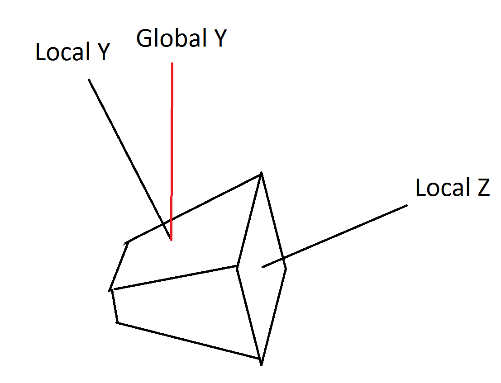
c. To rotate a part, you can press:  
 Q/ E: around local Y axis

d. You can also move the camera around to see in different position  
 U/ J: forward, backward  
 H/ K: left, right

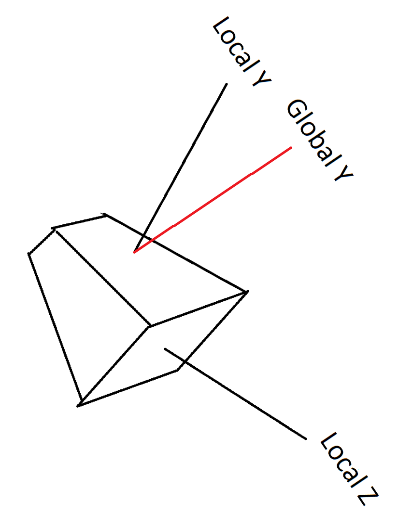
e. You can rotate the camera around to see in different angle  
 By holding the left mouse button and move the mouse around  
 Up/ Down Arrow: local X axis  
 Left/ Right Arrow: global Y axis, why?

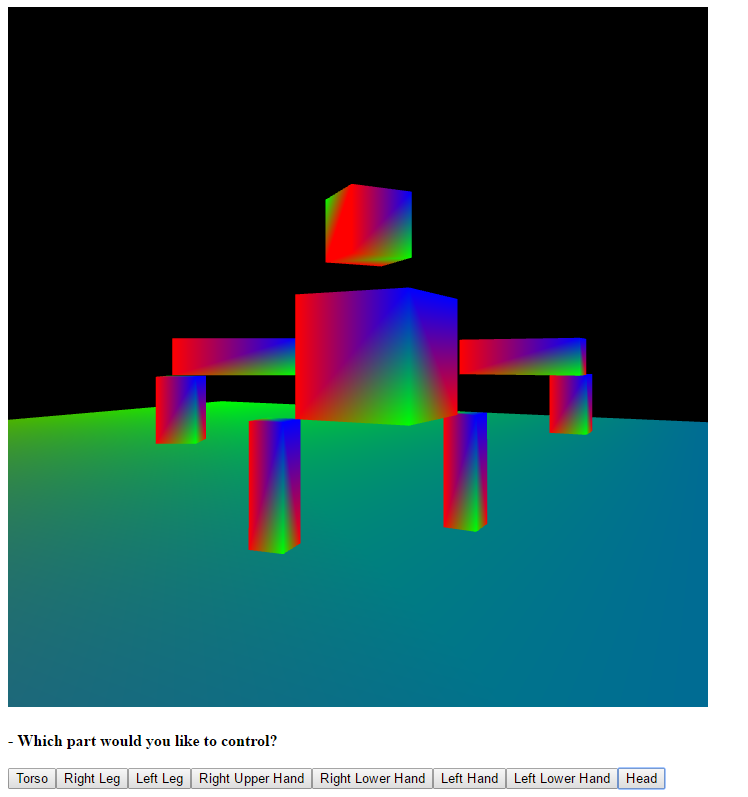


Considering a camera is facing upward instead of horizontally. If the camera is rotating along local Y axis 180 degree, it will face downward, which is not desirable. In other words, this is what we want after the rotation:



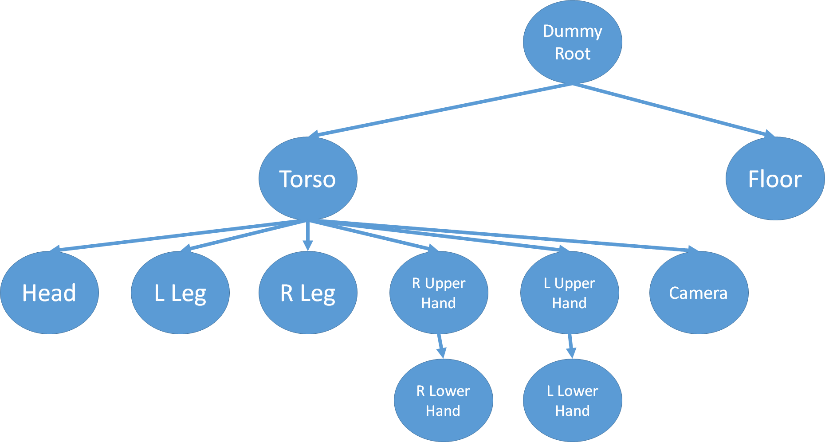
Instead of this:



After a sequence of operations, you can make the robot like this:

2. States which bonus tasks you attempted:

a. The whole scene is in 3D  
 b. A third person camera is implemented by attaching camera to the torso node of the robot  
 c. ● Although it cannot seen by user, I make a flexible structure for this lab, which can be reused for following lab assignments without too much effort:  
 ● Each object is a scene node, and they can attach children to form a scene node tree  
 ● By extending the class SceneNodes, I can render whatever I want. In this lab, I implemented class CuboidSceneNode to make the robot, and attach them together with ease  
 ● The class Scene holds the scene node tree, and maintains the class TransformStack  
 ● The scene node tree structure of this lab:



3. Lists which browser/OS you developed your code on (just in case)

Chrome/ Win10