

Project Report

22973676, Adrian Bedford

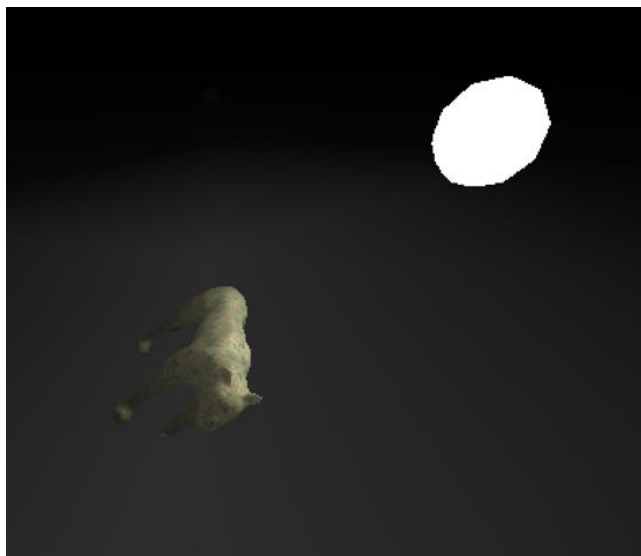
22989775, Oliver Lynch

Part A: Successfully implemented. Added scalars RotateX and RotateY onto Translate in the display() function. Translate is now multiplied by RotateY(camRotSidewaysDeg) and RotateX(camRotUpAndOverDeg). Added a second Translate() declaration to centre the camera orbit on the object, an optional improvement for viewing.

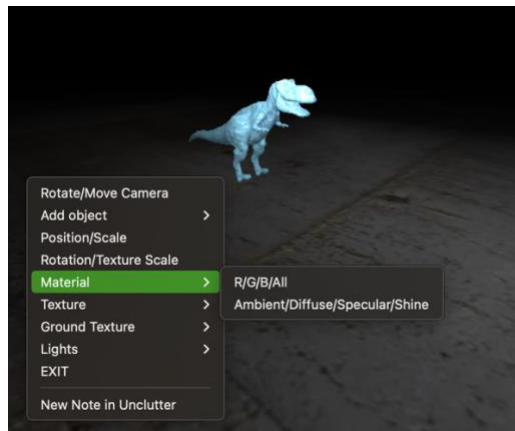


Part B: Successfully implemented. Added all 3 Rotate variables X, Y and Z, to the model matrix in drawMesh() and passed them each their respective angles from the scene object. To fix objects moving in the incorrect direction adjustAngleYX and adjustAngleZTexScale were changed, angle_yx[1] is now -= and az_ts[0] is also now -=, changed from +=.

Replaced constant value 2.0 within the texture2D parameter list inside the fragment shaders. It is now the texScale variable.



Part C: Successfully implemented. Created functions `adjustAmbientDiffuse` and `adjustSpecularShine`, then call these in `materialMenu` via id 20.



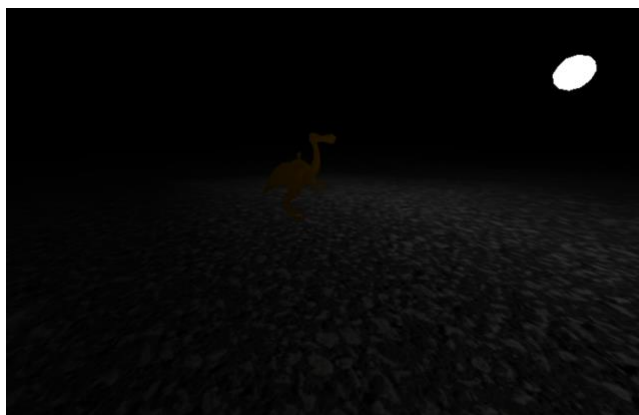
Part D: Successfully implemented. Reduced `nearDist` value to 0.01 instead of 0.2.



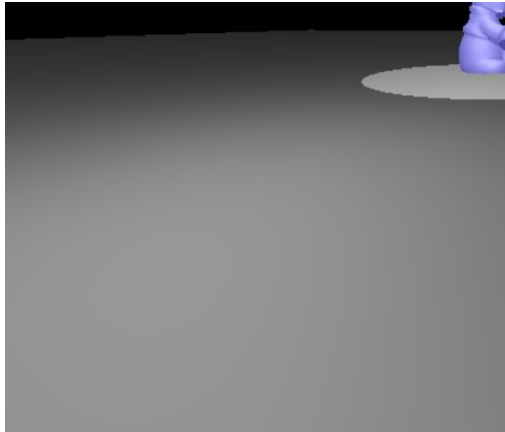
Part E: Successfully implemented. Added a second conditional to the original code in `reshape` under `Frustrum()`. The given conditional activates when the `width >= height`, but when the `width <= height` the bottom and top float are set to `nearDist * height / width`, which is swapped from original.



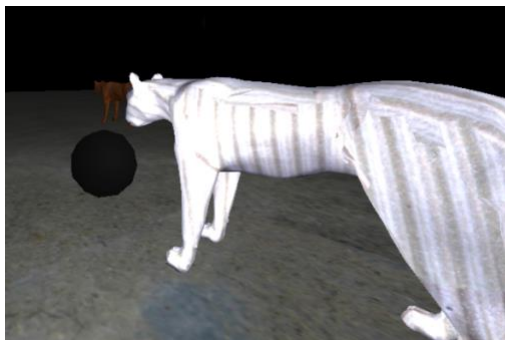
Part F: Successfully implemented. Added a falloff variable that calculates the inverse square of the distance between the object and light source, then colour.rgb is multiplied by this value. Located in fStart.



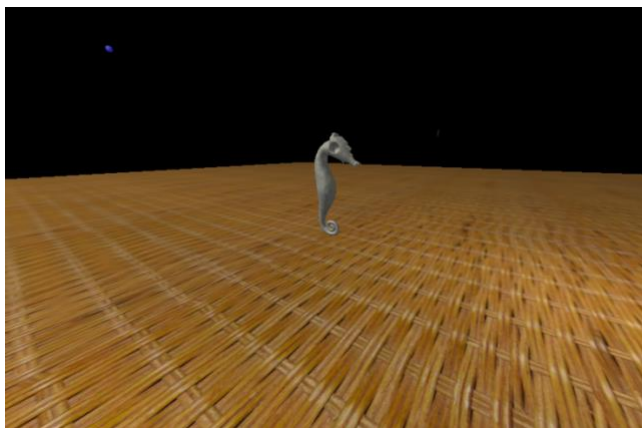
Part G: Successfully implemented. Moved all of the lighting calculations in the vertex shader into the fragment shader, and left only positional calculations in the vertex shader.



Part H: Successfully implemented. Separated specular by removing it from colour.rgb, creating a mix function in gl_FragColor and adding a length check to specular within that which tends colours towards black the bigger the value is. Specular always tends towards white.



Part I: Successfully implemented. Duplicated addObject(55) declaration in void init() and changed sceneObjs index to 2, also raised height of light source 2 slightly. Created the lightObj in display() as a SceneObject, and changed the for loop that applies Ambient, Difuse, Specular and Shininess to objects under the light source to account for the second source. Added id 80 to lightMenu() to be able to move the sphere, and added id 81 <= 84 to change its RGB as well. Duplicated the existing light calculations, then edited the light vector calculation in fEnd.



Part J: Successfully implemented. Added two cases to the keyboard() function, case 'd' and 'c', where d is delete and c is copy (duplicate). Case 'd' increments every index in the array from the object to be deleted, while 'c' adds an object of the currently modifiable object.

We developed a small selection system that allows you to traverse all current objects with the left and right arrow keys, it highlights the current object as blue. This was done in a function we created called select(), if left arrow is pressed the value of toolObj is decremented by 1, and up 1 for right arrow.

The spotlight can be moved and rotated around its centre, but looks a little odd. to do this we used the theta calculation available in David's google doc. Also, you can change its colour :).

