

# CSE 5542 LAB ASSIGNMENT 3

-Pavani Komati

1. (10%) Create a scene with multiple objects and/or instances of objects, along with a ground plane.

Created a scene with instances of two objects and a floor

2. (10%) Add at least one light source that casts shadows on the ground plane as well as on some of the objects.

Added a directional light casting on the objects and the floor. Textured.frag is used to implement the lightening effects for the model.

3. (20%) Create transformation matrices (aka a scene) to render the scene from the light source(s) and create a Shadow Map texture.

Maintained a depthViewStack and orthographicStack similar to the modelView and projection Stacks.

CreateShadowSceneGraph() creates the shadow scene from light position and saves to a shadowMapTexture.

4. (20%) Create a shader for use in shadow maps that is efficient (compared to lighting-based shader).

Shadow.vert and Shadow.frag takes care of creating shadow maps to the given objects.

5. (20%) Write a shader that adds shadow maps to the lighting and texturing of the objects.

ShadedTextured-vert.glsl takes care of finding the shadowCoord with the help of depthMVP matrix and

Textured.frag takes care of finding whether the pixel is in shadow or not.

```
vec4 lightPosInCamera = modelViewMatrix * lightPositions[1];
vec3 lightDir = normalize(vec3(lightPosInCamera) - ecPosition3);
float cosTheta = dot(normal, lightDir);
float visibility=1.0;
float bias = 0.005*tan(acos(cosTheta)); // to get smoother edges
bias = clamp(bias, 0, 0.01);
if(cosTheta>0 && (texture2D( shadowSampler, shadowCoord.xy ).z<shadowCoord.z-bias)){
    visibility=0.5;
}
```

6. (5%) Add viewport(s) along the right edge of the display of size at least 128 by 128. Display the shadow maps in these viewport(s).

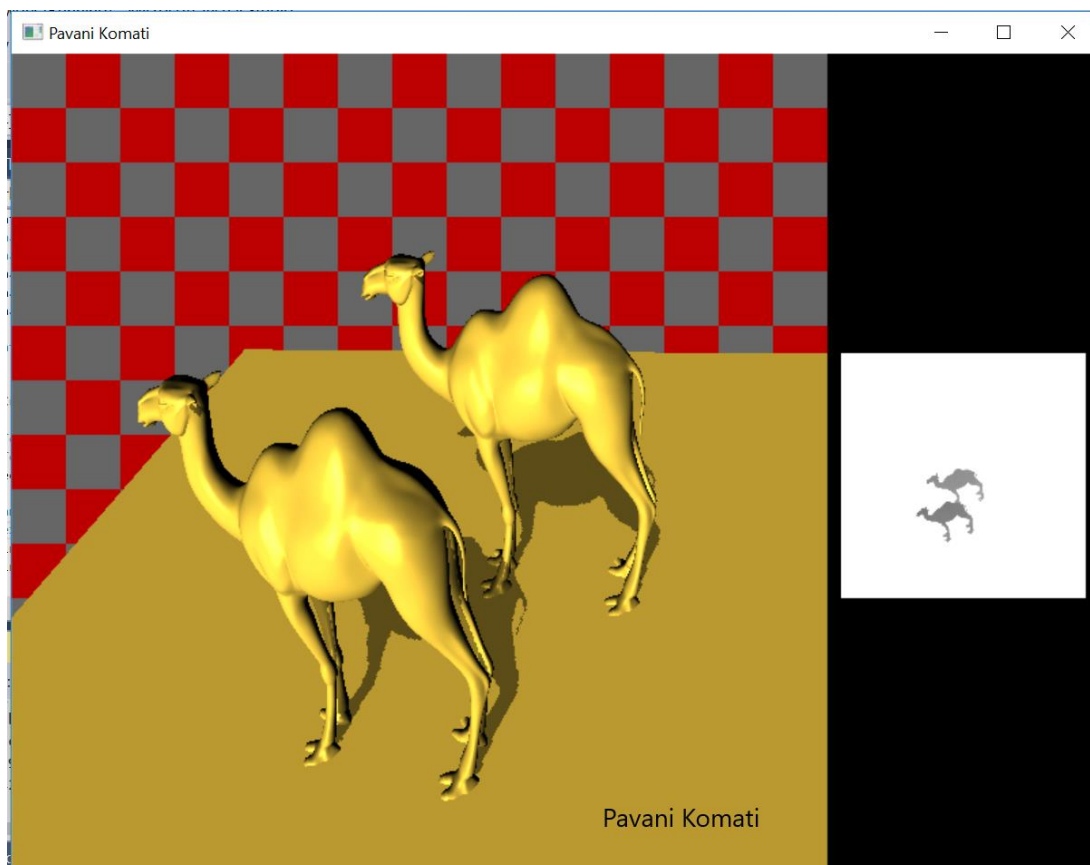
Displayed shadowMap by displaying the texture on a quad to the right.

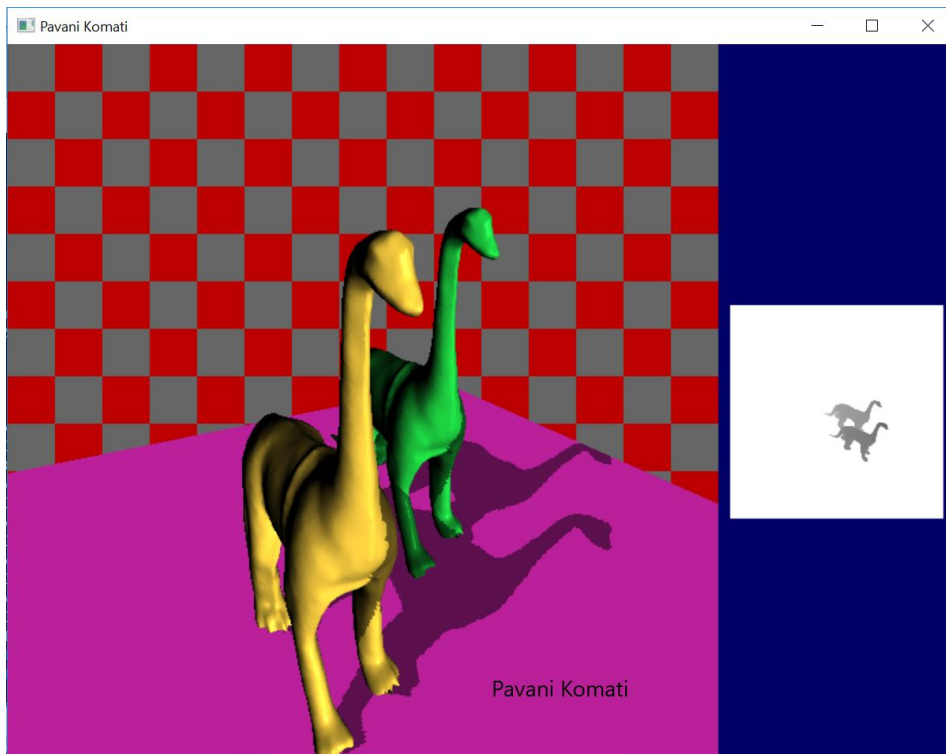
7. (10%) Add an ambient occlusion texture to at least one model and modify your shader to utilize it.

`AmbientOcclusionMapCamel1.png` Is used as the ambient occlusion texture .

```
vec4 ambient = vec4(lightAmbient, 1.0);  
vec4 occ=texture2D(OcclusionMap,uvCoords);  
if(occ.x>0)  
    ambient=ambient*occ;
```

8. (5%) Create 2 images show casing your work for the course web site. These images must have a pleasant backdrop and your name on them (use GIMP or photoshop). You will get zero credit for an image with a pure saturated color, black or white background. Points will be docked if your models use a pure saturated color.





9. Provide a readme.txt file for the grader that explains how to use your program.

Added instructions in the readme.txt in code

Extra Credit(5% each)

10. Attach an examiner to the light and toggle between the light and camera control with a keypress. This will give dynamic shadows. Note, you can use the same scene instance, but you cannot read and write to the shadowMap at the same time, so you have to change materials. The MaterialManager will let you do this.

Press 'l' (small L) to view the dynamic shadow  
Press 'c' to view the constant shadow

Constant shadows are created by saving the depthMVP matrices and using the same for changes to the scene.

Dynamic shadows are created by changing the depthMVP matrices according to the changes on the screen.

11. Add Variance shadow maps.

Please uncomment line 195 and comment previously declared shaders in Cube.cpp to use the variance shadow mapping.

```
192 VertexRoutine* vertexRoutine = new VertexRoutine("../Media\\Shaders\\ShadedTextured-vert.glsl");
193 FragmentRoutine* fragmentRoutine = new FragmentRoutine("../Media\\Shaders\\Textured.frag");
194
195 //UNCOMMENT FOR VARIANCE MAPPING
196 /*VertexRoutine* vertexRoutine = new VertexRoutine("../Media\\Shaders\\ShadedTexturedVar.glsl");
197 FragmentRoutine* fragmentRoutine = new FragmentRoutine("../Media\\Shaders\\TexturedVar.frag");*/
198
```

