**Math 155A - Introduction to Computer Graphics – Fall 2020
Instructor: Sam Buss,  Univ. of California, San Diego**

**Project #5 – Add Phong lighting to the plane, your initial & the surface of rotation.
Due date: Friday, November 20, 11:00pm.**

**Goals:** Continue working with your program from Project 4 to add Phong lighting to the scene. Use illumination and shading to make your scene from Project 4 look more three-dimensional. Learn how to shade objects with the Phong lighting model in OpenGL. Create four lights, including a spotlight, and create three materials.
**If your project 4 did not get completed successfully:** You may start the project using the floor and your initial. If necessary, please ask Professor Buss for code for the surface of revolution after Monday.

**Project 5, and the supplied source code, will be discussed in Monday’s zoom class meeting.**

**What to hand in:**

* **Upload to your shared google drive folder:**
	+ Your main new source files **MySurfaces.cpp, MyInitial.cpp, MySurfaces.h, MyInitial.h, PhongData.cpp and PhongData.h.** If there are changes to any other source files, please upload those as well.
	+ Also, **upload your PDF report** with (a) Report on consultations/feedback given and received. (b) Discussion of any special features or of any problems encountered, and how difficult or easy you found the project. (c) Two images, showing attractive views of your project.
* **Upload to gradescope: Your PDF report.**

**Get an individual zoom grading session** with Professor Buss or a TA.

**INSTRUCTIONS:**

1. **Make sure to save copies of all your source files from Project 4. This is very important!** They should also be saved in the Project 4 hand-in google drive sharedfolder.
2. **Download** the file [**PhongProject5.zip**](PhongProject5.zip)from the web. This zip file contains four C++ source and header files. It also contains an executable **PhongDemo.exe.**  The URL for this download is: file [**http://www.math.ucsd.edu/~sbuss/CourseWeb/Math155A\_2020Fall/Project\_5/PhongProject5.zip**](155A_Fall2020_project5.docx)
3. **Run the executable** on a PC (or CloudLabs). For this, first download the GLSL file EduPhong.glsl from the web at <http://www.math.ucsd.edu/~sbuss/MathCG2/OpenGLsoft/EduPhong/>; and place it in the same folder as PhongDemo.exe. When you run the program, you will see a scene with a surface of rotation and a letter "X". To get started: Use “**M**” to increase the mesh resolution. Press “**W**” to toggle wireframe mode. Then use “**P**” to use Phong interpolation instead of Gouraud interpolaton. The scene should now look decent (after switching from Gouraud interpolation to Phong interpolation. There are a lot of keystroke commands act on the scene. Some of them are familiar:
	1. Pressing the arrow keys changes the view position.
	2. Pressing the “**W**” or “**w**” key toggles wire frame mode.
	3. Pressing the “**C**” or “**c**” key toggles culling of back faces.
	4. Pressing the “**M**” or “**m**” key increases or decreases the mesh resolution on the spheres (ellipsoids), cylinders and tori.
	5. The animation can be turned off and on by pressing “**R**” or “**r”.** The animation can be made to run faster or slower by pressing “**X**” (faster) or “**x**” (slower).
4. There are also a number of new keystroke commands:
	1. “1” turns the first light off and on. There is a small emissive sphere that appears and disappears when the light is turned off and on. This emissive sphere is **\*not\*** the light: instead it just marks the position of the light.
	2. “2” turns the second light off and on.
	3. “3” turns the third light off and on.
	4. “4” turns a spotlight off and on.
	5. “P” toggles between Phong shading and Gouraud shading. (That is, between Phong interpolation and Gouraud interpolation.)
	6. “E” toggles emissive lighting off and on. (Note the spheres marking the lights appear and disappear, since they are the only emissive materials in the scene.)
	7. “A” toggles ambient lighting off and on.
	8. “D” toggles diffuse lighting off and on.
	9. “S” toggles specular lighting off and on.
	10. “H”/“h” selects using the halfway vector (“H”) and not using it (“h”)
5. Take some time to run the program and look carefully how things render in the Demo code. You should understand what all these keys are doing by the end of your programming project.
6. There is another source program available, **SimpleLightModern**, which is available at <http://www.math.ucsd.edu/~sbuss/MathCG2/OpenGLsoft/SimpleLightModern>.This program shows a light rotating around three spheres: the source code is available, and there is a web page at the EduPhong web page describing how to use the Phong lighting code.
7. Your job is to re-create the program PhongDemo.exe -- sort of!! You will use your initial and floor and circular mesh from Project 4 for this. (If your Project 4 code does not work, please email with Professor Buss for replacement code for Project 5.)
8. **Form a new project and solution; the next two steps describe how to integrate your project 4 code into the new system.** Include the all supplied source files in the new project. (Again, **save copies of your old source files!** Do not overwrite them.**)**
	1. There are six completely new C++ sources files and one new .glsl file. These are
		1. **PhongData.cpp** and **PhongData.h**. These plus MyInitial.cpp and MySurfaces.cpp are the primary programs you will modify. They set the light and material properties. These were in the zip file you downloaded
		2. **PhongProj.cpp and PhongProj.h.** These are the new main programs. They replace **SurfaceProj.h and SurfaceProg.h.** (The latter two files are no longer used.) These two files are in the zip file you downloaded.
		3. **EduPhong.cpp and EduPhong.h** C++ files for interfacing with the .glsl programs for Phong lighting. This is available from the EduPhong software web page at <http://www.math.ucsd.edu/~sbuss/MathCG2/OpenGLsoft/EduPhong/>.
		4. **EduPhong.glsl –** Shader programs that do the Phong lighting calculation. This replaces the old .glsl file. This is also available from the same EduPhong web page.
	2. Other than these files, please copy all of your Project 4 source files into your new Project 5: except do **not** include SurfaceProj.h and SurfaceProj.cpp. The files to keep include the C++ source and header files for GlGeom objects, GlLinear files, GlShaderMgr files and MathMisc.h, and your MyInitial files, and MySurfaces files.
	3. Create your Project 5 with Visual Studio. Include all the files in your project. Also, put a copy of the **EduPhong.glsl** file in the same folder as your Visual Studio project files (or in the correct directory for you computer set up).
	4. Your code will not compile yet. Follow the instructions in Step 9.
9. To get your program compiling.
	1. In both of your source files **MySurfaces.cpp** and **MyInitial.cpp** change #include “SurfaceProj.h” to #include “PhongProj.h”
	2. Decrease your variable animateIncrement to be much smaller, say 1/10th its current value, or perhaps even to zero. We want objects to move only very slowly.
	3. Everywhere the “vColor\_loc” variable is used, you will have a compile error. This will happen in both MyInitial.cpp and MySurfaces.cpp. Each line looks something like
	glVertexAttrib3f(vColor\_loc, 1.0f, 0.4f, 0.4f);
	You will replace this – for the moment --with the line
	myMaterials[0].LoadIntoShaders();
	This will make everything be colored (emissively) red. For your Project 5, you will change this red color to attractive Phong-illuminated similar to the colors you used in Project 5.
	4. You can start with using myMaterials[0],but for your programming project, the “0” will be replaced by some value 0, 1, 2, etc. so as to match the old color. You will probably want to use as many new “materials” in this project, as there were colors in your Project 4 (surface project).
	The PhongData.cpp file sets the value of myMaterials[i]. The supplied code makes the material purely emissive so it does not get used in Phong ambient, diffuse or specular light.
	5. You may need to add more materials if you have a lot of colors in your initial. If you have animated colors, a little more work is still needed to have animated colors in your Project 5 (this is optional: you may instead turn off the animation.)
	6. Once you have made all the above changes, you should have a running program that shows everything well in wireframe mode.
	7. These changes make the colors for the surfaces now be given as emissive Phong colors.
10. Your main programming task is to update **PhongData.cpp** and **MySurfaces.cpp** and **MyInitial.cpp**. Most of your work forProject 5 will be updating **PhongData.cpp.** Yours tasks are below:
11. **Create material properties.** You will design at least three or four different materials: One for the ground plane. One for the surface of revolution. And, at least one for the initial. These each need to have ambient, diffuse and specular colors, and a specular exponent. In the supplied code, the colors are purely emissive, with the other colors black (0.0, 0.0, 0.0). You will change these to be different colors --- they should be ***completely non-emissive***.
These materials are designed by changing the numbers in the routine **MySetupMaterials.**The supplied code comes with an array of three materials. You may wish to may need to add more materials if you have a lot of colors in your initial. If you have animated colors, a little more work is still needed (or you can turn off the animation.)
It is rare, but if your Project 4 initial does not have a good surface to apply Phong lighting to, you may add a sphere somewhere and use Phong lighting on it instead of your initial.
12. **Create three lights, by updating the routine** MySetupLights(). Values for the first light are already supplied, so you can see how it is done. You do not need to use the same values for the first light in your own project! It is suggested that you experiment with changing them. And, you need to design values for the second and third lights (that is, myLights[1] and myLights[2]). This is done by giving ambient, diffuse and specular values for each light. The suggested *x-y-*z positions for the lights are given in the array myLightPositions[3]. However, you may modify these positions somewhat if it improves your scene (e.g. to avoid hitting your initial). The positions are set in the routine LoadAllLights().
13. **Create a spotlight.** myLights[3] should be a spotlight. Set its emissive, diffuse and specular colors in MySetupLights(). Also, set its “cosine cutoff” value, and its spot exponent, in MySetupLights(). Set the position of the spotlight, and the direction of the spotlight in the routine LoadAllLights(). LoadAllLights() already has the needed code, but commented out. You will need to provide numbers for the position and the spot direction to make a nice looking spotlight.
14. **Suggestions:**
	1. You do NOT need to match the colors in PhongDemo.exe. In fact, you should choose your own colors and experiment with colors. It is probably best if the colors are similar to the ones you used in Project 4.
	2. Ambient and Diffuse colors are typically either equal or are scalar multiples of each other. However, it is OK to modify the ambient color if you want mimic the ambient light picking up some its color from the environment. **You may wish to scale down the Ambient colors to be less bright.**
	3. Specular colors on materials generally should be white or gray (that is, with R, G and B values all equal).
	4. Specular colors of lights should usually be a scalar multiple of their diffuse color. This way the specular highlights will be the correct color.
	5. Add material colors and lights one at a time to get the look you want in stages.
	6. A common problem is to set too many colors too bright. If you scene looks overly saturated or is “bright and washed out”, try lowering your color values. (This is a common problem in working this assignment.)
	7. Avoid using R,G,B color values for materials that are equal to zero. It makes the surface too monochromatic, and unresponsive to light in that color. However, it is OK for the ***lights*** to have a zero value for a color.
	8. **Use the “e”, “a”, “d” and “s”** commands to view ambient, diffuse and specular light separately. This will help you debug visual problems, and design your colors more effectively.
	9. **Try the “H”/”h” commands** to see your scene with the halfway vector computation, and without. Do you see any improvement of one method over the other?
	10. The spotlight will look much better when using Phong lighting. Point the spotlight at the floor, and circular mesh (or your initial) to make a nice image out of it.
	11. The EduPhong code supports the Schlick-Fresnel term, but you do **not** need to use it. (If you do try it, it should be on a surface that has a very low specular reflectivity, otherwise, you will probably not be able to see any effect from the Schlick-Fresnel term.)
15. **Goals for your scene:**
	1. Choose your own colors for the scene; do not just copy the colors of the Demo executable.
	2. Specular highlights should be visible for all three surfaces, at least under the right viewing conditions.
	3. One of the three lights should be mostly white to render the surfaces in their native colors.
	4. The lights and materials should give a really good impression of three dimensionality. (This is important!)
	5. There should be at least two or three ways to set the lights on and off to give an attractive scene. Choose your own colors, and try to make it a bit artistic color-wise.
	6. The scene will probably not look good in all combinations of lights, but you should try to avoid having it look terrible in any setting of the lights.
	7. Your code should still support all the keystrokes described above
16. **Upload to the shared google drive turn-in folder:**  The files listed at the beginning of the assignment along with any other changed source files.
17. **Upload to gradescope a single PDF file (one page, possibly two pages) showing two attractive images of your scene.**
18. **How to write the PDF report.** Call it “**Project5report.pdf**” (This name exactly, same capitalization!) It must have:
	1. **Your name and student ID on the first page, near the upper left corner.**
	2. **Feedback/consultation.** Next have a paragraph describing:
		1. Everyone you received feedback from, and the type of feedback they provided. If the feedback was helpful, describe this.
		2. Everyone you gave feedback to, and the type of feedback you provided.
		3. Feedback can happen either while you are writing your program (say, if you run into problems), or after your program is finished (say, if you just want someone to check that it looks OK).
		4. If you get feedback from someone not in the course, please explain.
		5. Your help from feedback should follow the guidelines for academic integrity on the course web: “*However, you should definitely NOT: hand in someone else's code as your own, directly copy code from others, or work too closely with one person for the entire quarter. These will be viewed as serious violations of academic integrity. It is OK however to see someone else's code, and then take a short break (say, three minutes) and then write your own version of the code on your own. If you are not sure what is permitted, please talk with a TA or Professor Buss. (Any help you get from a TA or Professor Buss is permitted of course!)*”
	3. **A short paragraph:** Describe any special issues that pertain to your project. State how difficult you found the project, or whether it was easy. Which parts were the most difficult?
	4. **Two pictures** showing attractive images of your scene. As in Project 4, the attractiveness of the scene will be part of the grade. However, here is to only the attractiveness and the design of the colors, illustrating your understanding of Phong lighting.
19. **Grading session:** Be able to show what is attractive about your scene. Which lights should be turned off or on to make the most attractive versions of the scene? Which lights can show specular highlights on which surfaces? Be able to discuss on Phong shading improves over Gouraud shading.

**Program grading: Scale of 0 to 10.** Personal grading session with a TA or Professor Sam Buss.