Student's ***NetID***______________Student's Name_____________________Grader's Name______________________
(netID == 3 letters, 3 digits: e.g. JET861 Please write clearly; make it easy to read)

CompSci 351-1 Grading Sheet: Project C   Fall 2019

J. Tumblin 11/15/2019

_______**10% Filenames, PDF report:** All file-naming correct (2pts) + clear illustrated PDF report (5pts) including name, netID, title, goals, user-guide, >= 4 results pictures, + correct sketch of your program's scene-graph showing all its transforms (3pts).

_______**5% On-screen User Instructions:** Program's on-screen display allows new users to quickly and easily identify and use all the programs features and options, without any extra help from source code, report, or authors' explanations.

_______**5% Ground-Plane Grid:** Project shows horizontal 'floor' or 'terrain' of repeated shapes or lines that extend nearly endlessly to all distant horizons, and thus let us easily assess changes to camera position and aiming direction.
In the world coordinate system where +z is 'up', the ground plane at z=0 spans x,y coords that appear horizontal on-screen.

_______**10% ≥3 Solid, Separate, Jointed, Continually Flexing Shapes:** Properly lit (Phong/Blinn-Phong) & shaded (Gouraud / Phong) 3D shapes at separate, different ground-plane locations, with continually-changing joint angles. Wireframe *not* acceptable!

_______**5% Large, Slowly-spinning Sphere** lets us visually confirm all lighting and shading methods: sphere is easily viewable from any direction and easily lit from any direction, and rotation reveals faceted/smooth effect of Gouraud/Phong shading.

_______**5% Single-Viewport Display fills entire browser window of any shape.** Re-sizing the browser window always keeps it filled with an undistorted image from a perspective camera with 30-degree vertical field-of-view; no shape distortions, no blank areas allowed except a fixed-height or fixed-width border & a region to hold HTML buttons, text, edit boxes, etc.; no browser 'slider bars'!

_______**5% Smoothly adjustable 3D View Control:** User interaction for unrestricted viewpoint control: be able to aim camera in any direction without changing position: be able to move forward/backward in the gaze direction, and 'strafe' sideways left/right from any 3D position; (HINT: 'glass cylinder' method; mouse or arrow-key aiming (tilt,yaw) and WASD to move fwd/rev, strafe left/right).

_______**10% Obviously different-looking Materials for each separate object**
HINT: use materials parameters given in starter code file `materials_Ayerdi.js`

_______**5% One 'headlight' light-source, co-located at camera eyepoint, that users can switch on/off**
(when correct, the specular highlights stay in the middle of any shiny sphere as camera moves)

_______**10% A 2nd light source at user-adjustable 3D world-space position, that users can switch on/off, (without affecting headlight) and with separate, user-adjustable R,G,B values for ambient, diffuse, and specular light amounts.**
Illumination on surfaces must NOT move when camera moves (though its specular highlight might).

_______**10% Interactive switching between all available lighting/shading methods** (requires at least two to earn this credit) without stopping or disrupting the program or its on-screen display.

_______**20% ≥Four lighting/shading methods:** Users can interactively select between Gouraud Shading and Phong Shading; for each of these, they can also select between Phong lighting and Blinn-Phong lighting; more methods welcome. Gouraud shading gives crudely-shaped highlights: Phong shading yields rounded highlights that can be smaller than triangles. Blinn-Phong lighting and Phong lighting yield slightly different specular highlights. (HINT: use different GLSL shaders for Gouraud and Phong shading)

EXTRA CREDIT:

_______2% extra credit: 3 or more user-selected distance dependencies (ATT) for your light sources:
(must include choice between NONE, 1/dist, and 1/dist^2, with dist calc'd at each vertex)
_______2% extra credit: geometric shape distortions in shaders, not reproducible by matrix transforms in Vertex Shader (e.g. twist vs. z; sinusoidal waviness, qualify, but simple scaling or displacement of selected vertices will not suffice)
_______2% extra credit: Advanced Shader Methods: in a visually obvious way, demonstrate Cook-Torrance or others such as 'toon' shaders that are not a sub-set of Phong or Blinn-Phong methods (see Lengyel book, search online).
_______2% extra credit: Simple Texture Maps (Chap 5-like; emissive only is OK)

_______2% extra credit per misc. feature (circle each item completed): ·3rd movable light-source, · user-switched materials for just one object (>10 visually distinct mat'l choices; no effect on other objects), Texture mapping; texture features (e.g. render-to-texture (a 'mirror', etc), bump-maps (Lengyel-like), use of texture RGB in specular term, as scaled xyz displacement, etc.)

______________**TOTAL POINTS/100**   (24% of final grade)