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)

EECS 351-1 Grading Sheet: Project C Win 2016

J. Tumblin 2/23/2016

_____**10% Report & User Instructions :** clear illustrated PDF file report with your name project title, goals, user-guide, code-guide, and example results.

_____**5% Ground-Plane Surface:** Draws all shapes on properly-oriented ground-plane: +z==UP.

_____**10% ≥3 Solid, Jointed, Continually Flexing Shapes:** At least 3 separate 3D shapes at different ground-plane locations, with continually-changing joint angles. Wireframe is *not* acceptable!

_____**10% Single-Viewport Display fills entire browser window of any shape.** Dragging the window's corner to change height & width always keeps entire browser filled with an undistorted image from a perspective camera with 40-degree vertical field-of-view; no shape distortions, no blank areas allowed except a fixed-height region to hold HTML buttons, text, edit boxes, etc.

_____**10% 5-DOF Camera Control: (move forward/back & sideways; pan left/right, tilt up/down)** Users can adjust views smoothly. One set of controls positions camera, others rotate it.

_____**10% Obviously different-looking Materials for each separate object**
HINT: use materials parameters listed in `materials_Ayerdi.js` (Week08 starter code)

_____**5% One 'headlight' light source attached to the camera that users can switch on/off**
(if it works, specular highlights stay in the middle of any shiny sphere as camera moves)

_____**10% One light source at user-adjustable 3D world-space position, that users can switch on/off, and separate, user-adjustable R,G,B values for ambient, diffuse, and specular light amounts.** Light must NOT move when camera moves; moving light should cause moving reflections.

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

_____**20% Four lighting/shading methods:** select between Gouraud Shading or Phong Shading for Phong Lighting or 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.

==========

_____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)

_____3% extra credit: geometric shape distortions in shaders, not reproducible by matrix transforms (e.g. twist, sinusoidal waviness, etc) implemented in Vertex Shader.

_____3% extra credit: Advanced shading: implement 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),

_____3% extra credit: Simple Texture Maps (Chap 5-like; emissive only.)

_____3% extra credit per feature: Advanced Texture Maps; render-to-texture (a 'mirror', etc)
(Lengyel-like: use texture RGB value as specular, as bump map, displacement, etc)

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