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)

CS 351-2 Intermediate Computer Graphics — Spring 2018
Project B: Your own Ray-Tracing Platform — (45% of final grade)

--Up to 85% of grade ---COMPULSORIES--- Your program *MUST* demonstrate:

_____10%: All file-naming correct + clear illustrated PDF report with name, netID,title, goals, how to get on-screen help, user-guide, ≥5 half-page pictures, lists the 'optionals' you chose for your project (below).
_____5%: Side-by-side Viewports (without disruption by any on-screen user instructions): WebGL on left, your ray tracer on right viewport (see starter code...)
_____5%: 3D ground-plane grid in world-space x,y plane, displayed in both WebGL preview & ray-tracing
_____5%: 3D interactive viewing: ability to aim stationary camera in any desired direction, move in the camera-aiming direction, and 'strafe' perpendicular to aiming direction(e.g.'glass tube'). Easy, intuitive, unrestricted 3D.
_____5-7%: User-adjustable antialiasing: 5% for these two user-selectable cases (+2% for more choices!)
1 sample per pixel with no jitter, and 4x4 jittered super-samples per pixel, and possibly more.
_____5-10%: **User control switches among 2 different 3D scenes, each with a ground plane-grid. (+5% more for 2 more distince 3D scenes). All scenes need:**
_____5%: Matched Shapes & Views: Camera view, pose, 3D shapes & positions match exactly on left & right.
_____5%: At least 3 spheres of different sizes, different positions, different materials in each of the scenes.
_____5%: At least 3 shapes of a third kind (neither ground-plane nor sphere); each at different positions & orientations ('different' != scaled or clipped sphere: show a cylinder, cube, torus, superquadric, or…?).
_____10%: At least 2 individually user-switchable (on/off) point-light sources that cast overlapped shadows that combine properly in the ray-traced image. (No shadows in WebGL preview: too difficult!).
_____5-7%%: At least 2 of these light sources have independent, user-adjustable world-space 3D positions; you may attach one light to the user-positioned camera to make a car 'headlight' if you wish.
(+2% for proof of exactly-matched light positions in BOTH ray-traced image and the WebGL preview).
_____ 5%: In WebGL preview, show at least 2 different Phong materials on different surfaces simultaneously.
_____ 5%: In ray-traced result, show at least 2 different Phong materials on different surfaces simultaneously.
_____10%: Show easily-visible recursive mirror-like reflections between at least 2 different objects.
_____ 5%: User-adjustable recursion depth for rays: render 0,1,2,3,… 8 inter-reflections for all scenes.

--*At Least 15% of grade ---OPTIONALS--- Your program MAY demonstrate:*

_____+5% to demonstrate a non-uniform shape distortion using the worldRay2Model matrix or more; can you 'squeeze' a sphere to a flat pancake-like shape, *etc*? What does a 'perspective' transform do to a cube?
_____+5% for each new geometric shape type (past plane, sphere, and 3rd kind): Cube, Cone, Cylinder, Torus, Tetrahedron, super-quadrics, Blinn Blobbies or Meta-Balls, Bloomenthals' "unchained geometry", *etc*.
_____+5%: for each different kind of 3D procedural material implemented: 3D checkerboard, 3D cell textures (3D hexagons? Penrose tiling?), Bump mapping, Perlin Noise, Fractal textures, Turbulence fcns, *etc*.
_____+10%: for each published non-Phong light/material model implemented: (e.g. BRDFs, BSSRDFs, reflectance models by Cook-Torrance, LaFortune, He, Matusik/McMillan, Torrance-Sparrow, …)
_____+10%: for each CSG operator {(+)Union, (-)Difference, (*)Intersection}implemented & demonstrated by sphere, cylinder, cone, or other curved shape types. (No CSG on WebGL shapes--too hard!)
_____ +5%: One or more Area Light Source that casts soft shadows with no steps or hard edges. Example: rectangle light source similar to overhead fluorescents. (use jittered super-sampling on surface)
_____+10%: Transparency/Refraction governed by Snell's law, with exponential 'extinction' (loss of light vs. distance through material). Create a curved 'glass' shape and show how it distorts scene behind it.
_____+10%: Root-finder for non-analytic ray/object intersection finding: e.g. Newton-Raphson method, conjugate gradient, *etc*. Demonstrate on Blinn Blobby objects or other simple exotica.
_____+10%: implement ray-traced texture-maps or environment-maps method that apply photos to ray surfaces

_____/100 TOTAL
5/22/2018 8:59 AM