

# Laboratory #2: 3D Transforms & Lighting (坦克 + 黃色小鴨)

- Draw a 3D ground plane – it can be as simple as a flat grid of squares.
- Draw a tank consisting of a **body**, **turret** 砲塔, and **gun**. The geometry of these parts can be as simple as a box and a couple cylinders.
- (**Optional**) Texture mapping of terrain and tank.

# Laboratory #2 (2)

- Use **hierarchical** transformations to draw the body at a 3D tank location, the turret in a specific location relative to the body, and the gun in a specific location relative to the turret.
- Furthermore, draw a rubber duck 黃色小鴨 sitting on a specific location relative to the body, or relative to the turret, if you like.

# Laboratory #2 (3)

- Allow the user to control, with keyboard and/or mouse, the placement and rotation of the body, the **left-right** rotation of the turret, and the **top-down** rotation of the gun.
- Make head-to-head with **multiple** tanks (pretty easy, except for good key handling), and let tanks shoot bullets.

# Laboratory #2 (4)

- Have an initial camera location and look vector such that the tanks and ground plane are in view. Use ***gluLookAt()*** to specify these.
- Use a perspective projection (instead of, say, the orthographic projection).

# Laboratory #2 (5)

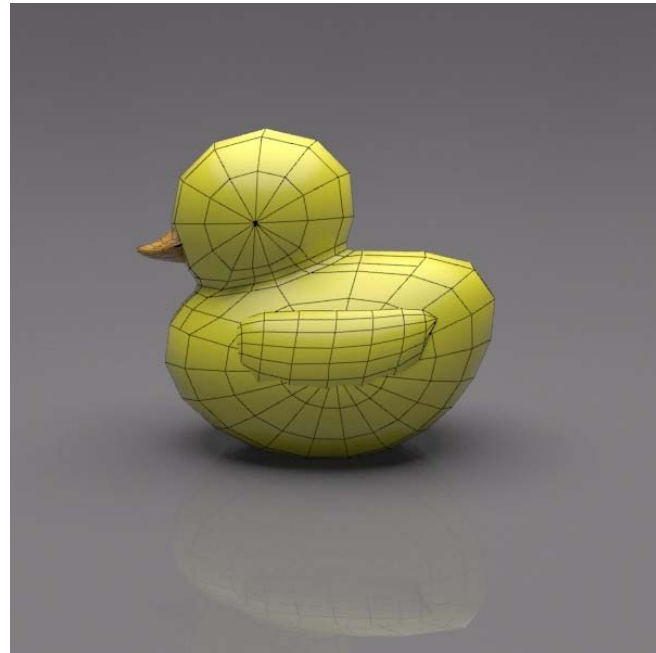
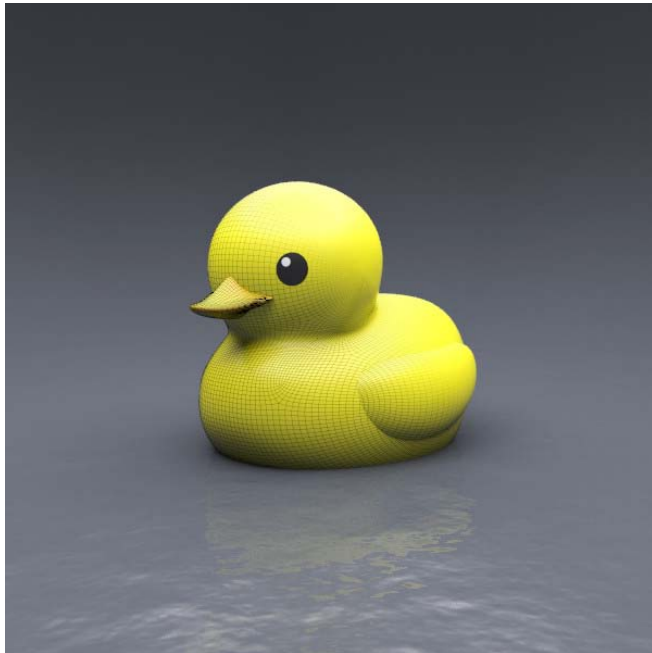
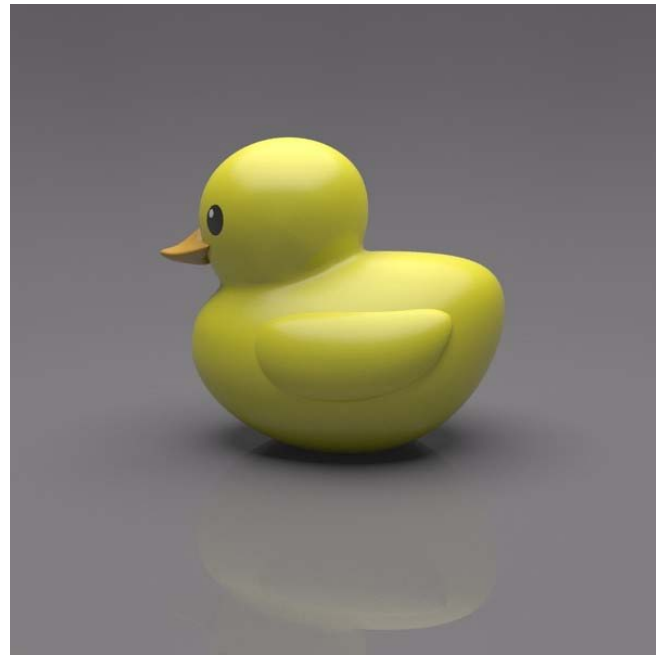
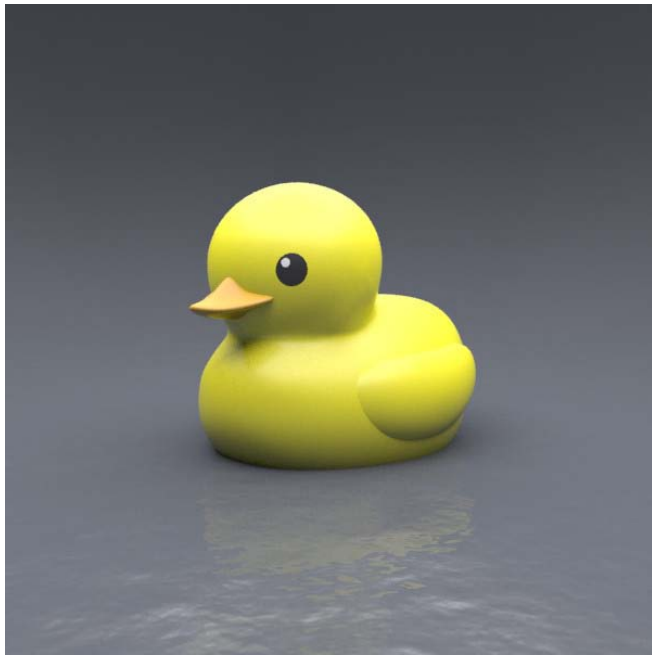
- Allow the user to control the 3D location and rotation of the camera.
- First-person mode: toggle back and forth between **third**- and **first**-person views.

# Laboratory #2 (6)

- (Optional) Automatic camera placement: design a camera algorithm that moves the camera in order to keep the tank(s) in view.
- Be careful with the near and far clip planes!

# Laboratory #2 (7)

- Add a **point** light to the scene and render with lighting.
- A menu that allows the color of such light source to be selected from white, red, green, or blue.
- Consider how to properly **pan** and **tilt** the light source, in particular the position should remain fixed but the direction will need to be adjusted.





# Laboratory #2 (8)

- Moreover, you will be given a program that loads into memory polygonal models, e.g., a rubber duck 黃色小鴨, in the **smf** file format. In addition to the **smf** model(s), add additional objects, if you like, to create an environment.
- Add OpenGL Illumination to display your objects. To do this, all vertices need to have their normals defined.
- Average the normals for each face to create a normal for each vertex.

SMF is a subset of the Wavefront OBJ file format.

# Laboratory #2 (9)



黃色小鴨實為立體；此為示意圖

[Movie in action](#)

# Laboratory #2 (10)



黃色小鴨實為立體；此為示意圖



# Laboratory #2 (11)



黃色小鴨實為立體；此為示意圖