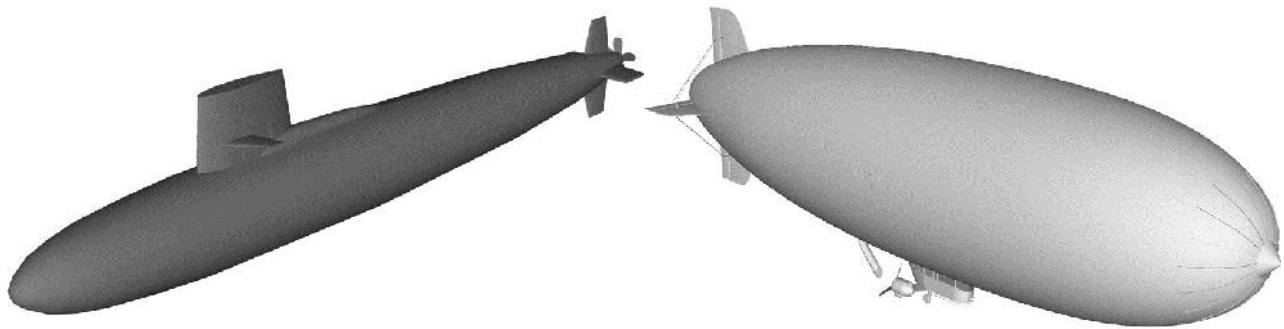


# CPS511 Assignment 1:

## Using Geometric Transformations to Construct and Manipulate Multi-Part Models

**Due Date: Fri., Oct. 14 11:59pm**



You will construct and interactively manipulate a complex multi-part model. This programming assignment will increase your knowledge of geometric transformations, shape modeling, and simple animation. **You must do this assignment alone - no groups. Do not attempt to find source code on the web for this assignment. It will not help you and you risk extremely serious consequences.** Begin designing and programming early! This project is worth 15 percent of your mark. **If there is some part of the assignment you do not understand, please see me (or email me) as soon as possible and I will clarify the issue.**

### **Program Functionality Requirements:**

You may use the skeleton code provided with the assignment. This code sets up lighting and shading for you, creates and draws a quad mesh for you to act as the ground. The code also provides a simple cube mesh for you to use in this assignment or in future assignments. In the skeleton code, it is currently unused. The skeleton code also sets up the material for your submarine model and uses simple transformations of a cube, created by calling `glutSolidCube()`, to position the cube.

Your program should use OpenGL transformations (`glTranslate`, `glScale`, `glRotate`) to construct a submarine model (or a dirigible model - your choice) as in the figures above. Your model does not have to look sophisticated - you are not marked on how good your model looks.

**You may use your own meshes for the submarine parts as long as you are positioning them using the OpenGL transformation functions.** However, it is sufficient to use the glut shape primitives to draw the parts:

- `glutSolidCube(size);`
- `glutSolidSphere (r, nLongitudes, nLatitudes);`
- `glutSolidCone(rBase, height, nLong, nLat);`

- `glutSolidTorus(rCrossSection, rAxial, nConcentric, nRadial);`

and/or the glu quadric primitives:

- `gluSphere()`
- `gluCylinder()`
- `gluDisk()`
- `gluPartialDisk()`

Keep in mind that the quadric surfaces have to be created using glu functions. The following lines of code are an example of how to use them:

```
1. GLUquadricObj *mySphere;
2. mySphere = gluNewQuadric();
3. gluQuadricStyle (mySphere, GLU_LINE);
4. gluSphere (mySphere, r, nLong, nLat);
```

This will be explained in more detail in an upcoming lecture. You can google these functions to get details on the parameters.

**NOTE: You must use `glTranslate`, `glRotate`, `glScale`, and `glPushMatrix()` and `glPopMatrix` to implement all transformations!!**

## Requirement 1:

Your submarine (or dirigible) must consist of at least 5 parts. You must have a working propeller that spins as the submarine (or dirigible) moves forward. Use `glTranslate`, `glRotate`, and `glScale` to position the parts and construct the sub.

## Requirement 2:

- 1) You must use the keys (or the mouse) to control the forward movement of the sub. Use 2 keys to control the forward speed of the sub, where the speed ranges from 0 to a predefined maximum. Use `glTranslate()` to implement forward movement. The faster the sub goes, the faster the propeller spins.
- 2) Use 2 keys (I suggest the left and right arrow keys) to control incrementally turning the sub right or left. Use `glRotate()` to implement turning.
- 3) Use 2 keys to control vertically raising or lowering the sub (I suggest the up and down arrow keys). There is no requirement pitching the sub on an angle when raising or lowering - the sub may remain horizontal.

## Requirement 3:

Implement a "help" key that when pressed tells the user which keys are used and their function.

## Note:

1. You do not need to provide any navigation (camera view change) capabilities - you may use a fixed camera view.
2. You do not need to provide a window reshape capability.

## Optional Bonus (1 mark)

- 1) Use an indexed VBO to draw your ground mesh (0.5 marks).
- 2) For advanced students: use a vertex and a fragment shader for rendering (0.5 marks).

The maximum bonus for the assignment is 1 mark.

## Grading (Out of 15 marks)

Multipart submarine (or dirigible) consisting of at least 5 parts. Must have a working propeller. Must use <code>glTranslate</code> , <code>glRotate</code> , <code>glScale</code> to perform transformations of submarine parts.	7 marks
Simple animation via <code>glTranslate</code> and <code>Timer</code> function to move sub forward. Use keys (or mouse) to control forward <b>speed</b> of sub, left and right turn control, and up and down control.	7 marks
Help key to explain interface	1 mark
Bonus	1 marks
Total	16 marks

## Program Submission

Use D2L to submit your assignment. Submit all your source files. You may use C, C++, or java for your program. **Zip everything up into one file.** **Do not include executable files!** If your program runs under Windows, include a README file describing how to compile your program. If you want to inform the TA about your program (special features, bonus work etc.) include this information in program comments and the README file. If only part of your program works, list the parts that work and that do not work in the README file. Include all makefiles (for Linux) or solution/project files (for example, if you used Visual Studio). **It is your responsibility to ensure the TA has enough information so that he can, with little effort, compile and run your program.** I am being flexible in terms of your programming language and operating systems choice so you must make an effort to meet me halfway. If the TA has trouble compiling your program, he will have the discretion to deduct marks and/or he will ask you to compile and run your program in his presence.

