3D Transformations

CS 452 - Computer Graphics
Assignment 3

We want to design a mobile robot that can perform the task of drilling. The design of the robot is shown in the figure below. The robot consists of a drilling needle attached to its arm. The arm is mounted on a rotating base which is mounted at the middle of a moving platform.

1. Your robot has to have at least the following elements:
   1. Wheels
   2. Main Platform
   3. Circular revolving base
   4. Lower Arm
   5. Upper Arm
   6. Needle

2. For each of these parts, you will have to create one (and only one) function draw(), for example: drawWheel(), drawPlatform (), drawBase(), etc. You can draw these parts in a very simple way, for instance only in wireframe (just drawing few segments for each part), or using some primitives provided by glut.

3. You will have to use transformation matrices to correctly place the robot parts. For example, you will use a transformation to correctly place the rear wheel frame before calling drawWheel(), and then a new transformation will be set for the front wheel frame, and the same function drawWheel() will be called again.

Transformations

First on a piece of paper complete the OpenGL code given below to simulate the forward kinematics of the robot. Take center of the platform to be origin. Assume that a draw function is already provided that can draw the vertices for each of the parts, such as drawWheel() to draw a wheel and drawBase() to draw the revolving base of the robot etc.

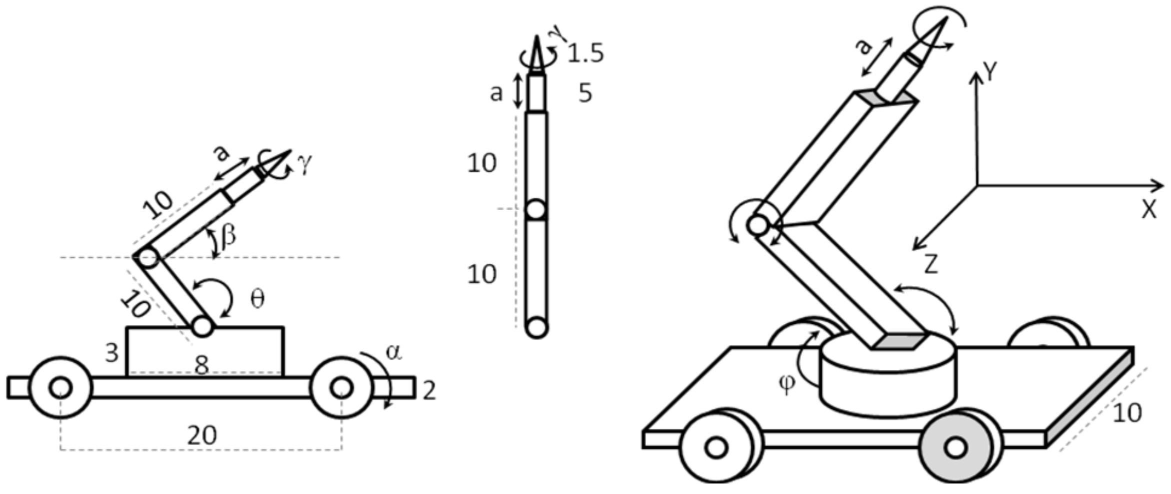

Figure 1: Schematic of Mobile Drilling Robot

```
…
Translate(7,0,0)          //translation of 7 along x-axis – can also use T_x(7)
drawBody()
PushMatrix()              //transformations for shaded wheel
    Translate(10,0,5)     //translation of 10 along x and 5 along z axis
    Rotate(-α,0,0,1)      //rotation of – α along z-axis – can also use R_z(–α)
    drawWheel()
PopMatrix()
```

After you have figured out all the transformation matrices and their ordering, figure out all the transformations required to create each component of the robot separately at origin. You need to perform this for each draw function.

Implementation in OpenGL

Finally code all these transformation in OpenGL. Skeleton code is available in MobileRobot.c file which can be downloaded from LMS. Either create Visual Studio project using this file following the instructions mentioned in OpenGL Handout or compile the file directly using msys/mingw console (msys/mingw console can also be downloaded from LMS). You are suggested to use glutWireCube, glutWireCone and glyCylinder in the beginning and once you are done with your drawing and transformations you can replace them with glutSolidCube with glutSolidCone.

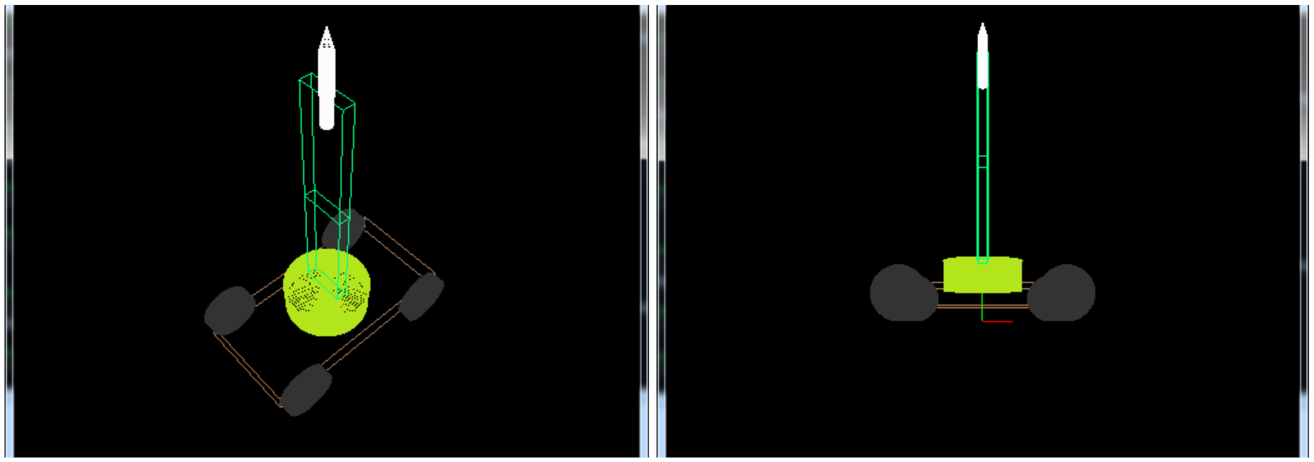
Figure 2: Mobile Robot with all six parameters having zero value. This output is without lighting and shading and is using wire objects for better visibility. Also note a small xyz axes.

Lighting and Shading

You are also required to assign appropriate material color to each component. This can be done by adding the following lines at the beginning of each of your draw function.

```
GLfloat color[] = {0.2,0.8,0.2};
glMaterialfv(GL_FRONT, GL_AMBIENT, color);
```

See drawTerrain() function as an examples.

Rest of the code for lighting and shading is already present in the skeleton code. To enable lighting and shading in your project you have to uncomment last 2 lines of function void initLight(void)

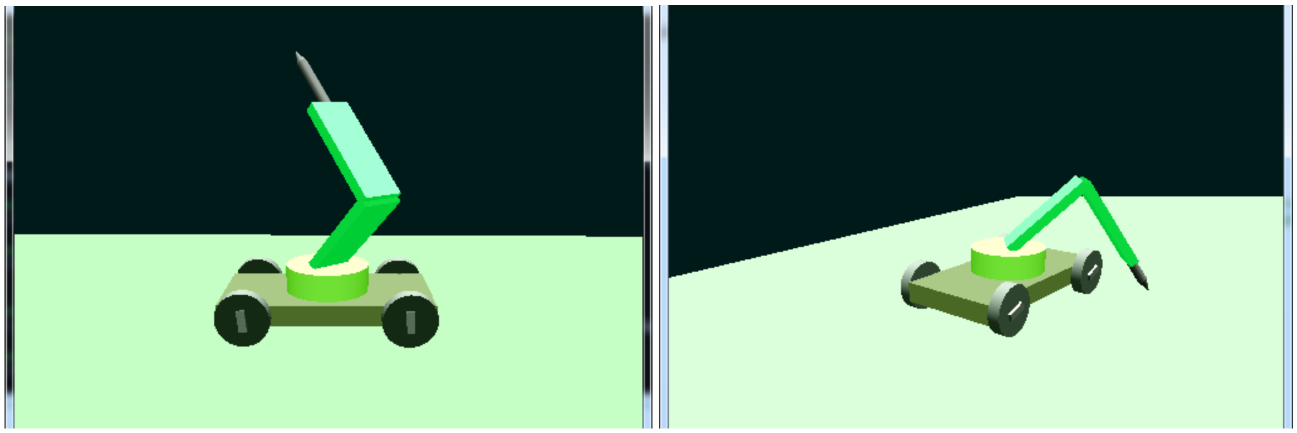
Figure 3: Final desired output showing our robot at work

Keyboard Controls

The user should be able to change the values of all the 6 parameters mentioned in the Figure 1 using inputs from keyboard. Keyboard functionality is already implemented in skeleton code for camera movements only. You are required to add keyboard functionalities for the remaining 6 parameters.

For more details download the **MobileRobot.exe** from LMS. You can use following keys to operate various controls in the exe:

- q – rotate cameras along x-axis in clockwise direction
- z – rotate cameras along x-axis in anticlockwise direction
- a – rotate cameras along y-axis in clockwise direction
- d – rotate cameras along y-axis in anticlockwise direction
- w – rotate cameras along z-axis in clockwise direction
- x – rotate cameras along z-axis in anticlockwise direction
- f and Shift+F – α
- b and Shift+B - β
- p and Shift+P - ϕ
- g and Shift+G - γ
- n and Shift+N – a
- t and Shift+T - θ

Submission Details

- DUE: Sunday Oct 18th, 11:55 p.m.
- Submission will be done on LMS.
- You must submit only your MobileRobot.cpp file. In case you have developed your code using MSYS/MINGW, submit all the files you have modified along with instruction on how to compile the code
- Submit at a small video of your project demonstrating movements of your robot