In this assignment you will be adding a robot conductor to your virtual train set. The robot can stand in the center of the track and will be composed of moving parts, specifically a rotating head and arms. Initially, the head is looking forward and the arms are at the side (as shown in first diagram) and the train is stationary. When the user starts the animation (from a menu item labeled “Animate”) a sequence of animations take place: 1. Robot elevates one arm such that it’s straight out (as shown in diagram 2). 2. When the arm reaches this position, the train starts moving at a slow speed. 3. After at least two turns around the track, the robot moves his arms such that the lower arm is raised (as shown in diagram 3). 4. When the arm reaches this position, the train moves at a faster speed. 5. After at least two turns around the track, the robot lowers his arm back to the initial position (in first diagram). 6. When the arm is in this position the train stops. 7. Have the robot rotate his head from one side to the other (180 degrees around the y-axis). The whole animation should restart if the user selects “Animate” again.
Part 1: Creating and animating the robot and train. (75 points total)

1. Robot model. (10 points)
The robot is composed of a sphere with two circles for eyes (so that the rotation can be distinguished), a cylinder for the torso, two cylinders for each arm and a cylinder for each leg. Each arm is composed of an upper and lower arm. The head is positioned such that its on top and center of the torso, the arms are at the top left and right of the torso. The legs are at the bottom left and right of the torso. Since the legs are not animated, each can be a single cylinder.

2. Animating the robot (40pts)
Using the diagrams above, the head should rotate around the y-axis from one side to the other which means a maximum of 180 degrees. The arms should rotate as follows: the upper arm should rotate at the shoulder about the z-axis; the lower arm should rotate at the elbow about either the x-axis or the y-axis depending on the motion sequence.

3. Train animation (25 points)
To change the speed of the train, use a timer function callback available in glut (instead of glutIdleFunc()). The relevant registration function is:

```c
void glutTimerFunc(unsigned int msecs, void (*func)(int value), value);
```

The callback description is below:

`glutTimerFunc` registers the timer callback `func` to be triggered in at least `msecs` milliseconds. The `value` parameter to the timer callback will be the value of the `value` parameter to `glutTimerFunc`. Multiple timer callbacks at same or differing times may be registered simultaneously. The number of milliseconds is a lower bound on the time before the callback is generated. GLUT attempts to deliver the timer callback as soon as possible after the expiration of the callback's time interval. There is no support for canceling a registered callback. Instead, ignore a callback based on its `value` parameter when it is triggered.

Extra credit: Applause

1. Add hands and make the robot clap (15 points): Add hands at the ends of the lower arms. At the end of the above animation, make the robot clap. (any type of motion that looks like clapping is acceptable).
2. **Texture map ground plane (10 points):** Add a ground plane to your model (it’s a plane that your tracks and robot stand on) and texture map the ground plane with an image. Code for importing ppm images into your program will be available on the web site.

Note: if you are planning to do the extra credit, ensure you first have a working version of the required assignment. Save the working version in a separate project and then embark upon the extra credit. If you can’t demonstrate a working version of the required part, you will not get full credit for it.

Submission:
1. Report explaining the algorithm, description of functions, and any other implementation details that explain your code.
2. Entire Visual C++ project directory including source files, header files and the compiled executable.

Submission process: Login to your webct account at webct.ucsd.edu. Click on Student presentations, and click on the “Edit files” link by your folder. Create a new subfolder in your folder called “Assign4”. Upload a zipped up file into this new folder.

Late penalty will be applied to assignments turned in after the time stated above.