Challenge One: *Java/Ego*

ROBOTIC AUTONOMY
Summer 2002

This week you are creating your TrikeBots: building them from scratch and then dressing one to represent your team for the next few weeks. But there are two critical activities that you must also complete to fulfill the Challenge. These relate to you, both individually and as a team, learning how to use our Java programming environment. So, we have two simple Java assignments for you so that you can get in there and start playing.

Java

Program 1: Calculator (10 points)
Execute a java application and demonstrate the completed Calculator tutorial we have provided for you. You will be graded based on your completion of this simple tutorial. We will test this by adding two numbers using your program. This will be graded separately for every student in class. **All other grades are awarded according to each teamís performance.**

Program 2: Timer (10 points)
Implement two buttons: START and STOP. When you click START, then later click STOP, your program should display the number of seconds that passed between clicking START and STOP. This should work repeatedly. We will grade this assignment by having you launch your application, then we will push the buttons and verify that you are printing the correct timer output. This will be graded by team, so the team should present their one solution to the assignment. (*Hint: the UserThread.java is instantiated anew every time you click a button. So do not put any global variables there, put them in UserWindow! This is very important.)*

Ego (30 points)

We all need a snappy outfit for our ego, so why leave your TrikeBot out? You must make your TrikeBot unique! Itís important so that you never, ever lose your TrikeBot in the masses of TrikeBots living in Building 17. Your TrikeBot should represent the unique personality of your team and no other. Robot fashions change just like human fashion, so donít dress your TrikeBot to look forever the same.

Itís not all about fashion. Youíll have to design your TrikeBotís outfit to satisfy several key constraints related to its future successful operation. Your modifications:

- should not interfere with the robotís motion and servos.
- should not interfere with CMUcamís field of view (FOV).
- should shield CMUcam from glare outside its FOV.
- should make your ëbot recognizable from 360 degrees for an observer 2 meters away.
- must pass our ìdumb-dumb testî ñ it canít make the TrikeBot inoperable. They also need to pass our ìtaste testî ñ it canít make the TrikeBot offensive.

We will grade the Ego challenge by first verifying that you meet the above constraints (20 pts), and then by judging the quality of your design. (10 pts)

Web Documentation (50 points)

On Wednesday you will deliver material for your first team web site. To get you in the habit of providing detailed project implementations, give us some description about how and why you made your Java programs and your Trikebot design. Questions to discuss include:

- What errors came up in your programs? What did you do to debug them?
- What compromises did you have to make in your TrikeBot design? How did you choose the materials to use?

Treat this first website as a chance to show off your TrikeBotís design and your team name. At your suggestion, weíll document your design, so tell us the best angles and close-up shots of your TrikeBotís unique characteristics and weíll snap away.