
1.124J Foundations of Software Engineering

Problem Set 7

Due Date: Thursday 11/16/00

Reference Readings: From [Java Tutorial](#)

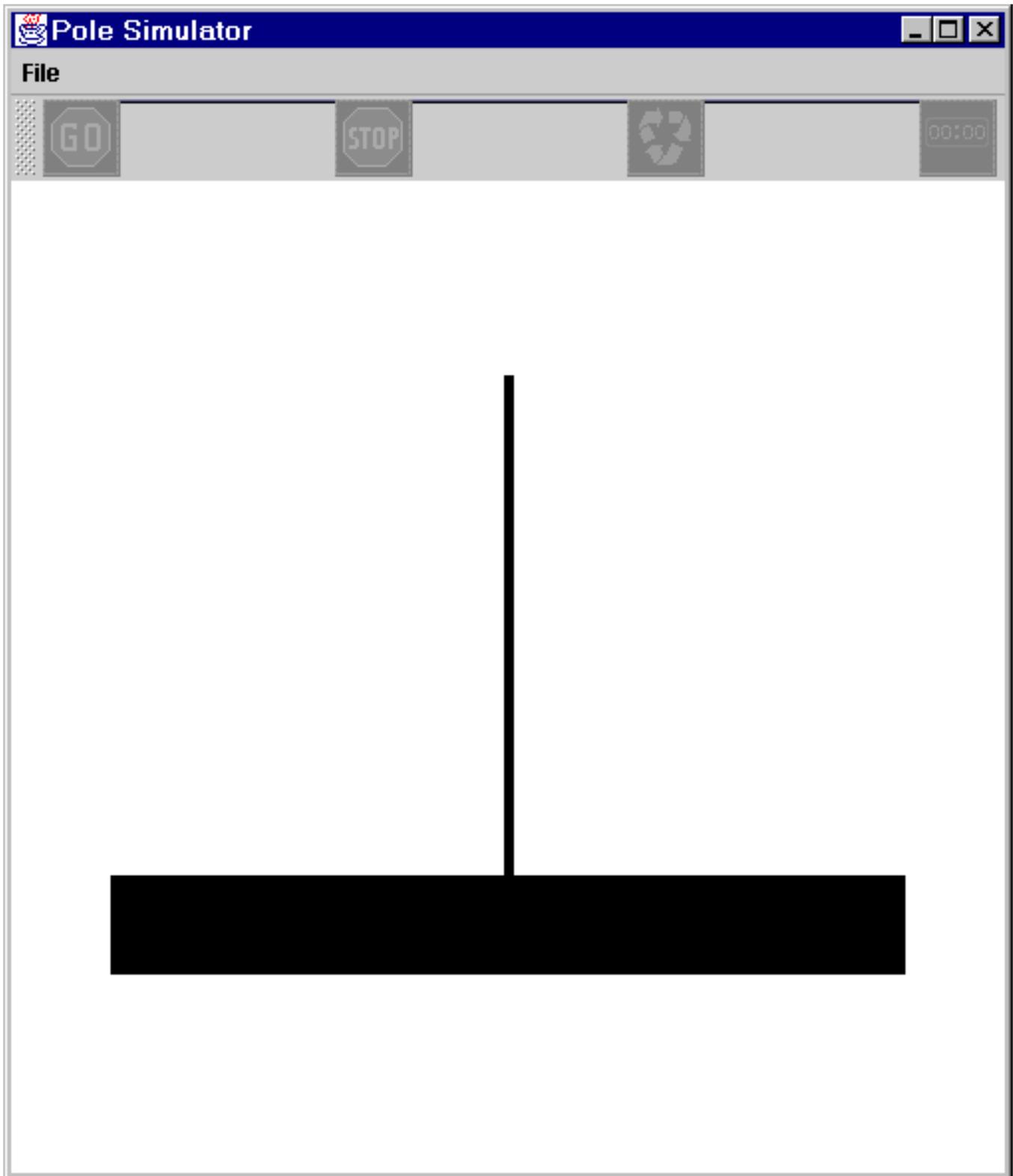
- *All material from the previous problem sets*

Problem 1: [100%]

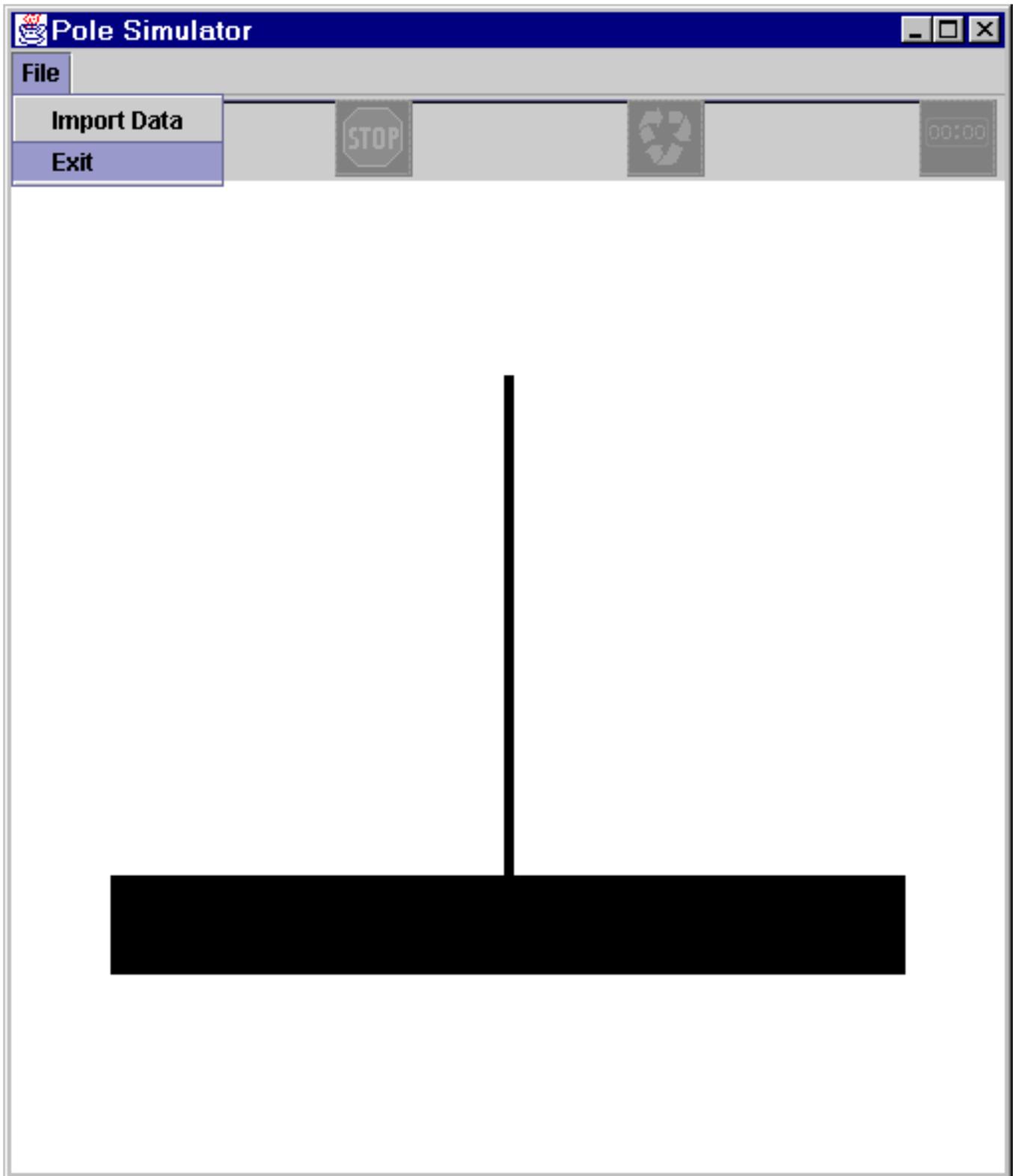
In this problem set you need to write a Java application which would simulate a moving pole.

The program should be able to read values of displacements from the pole data file and then allow the user to simulate the motion of the pole as it moves according to the displacements provided in the *pole.data* file. The file has values of times and displacements.

Initially the program should look as the one shown below:

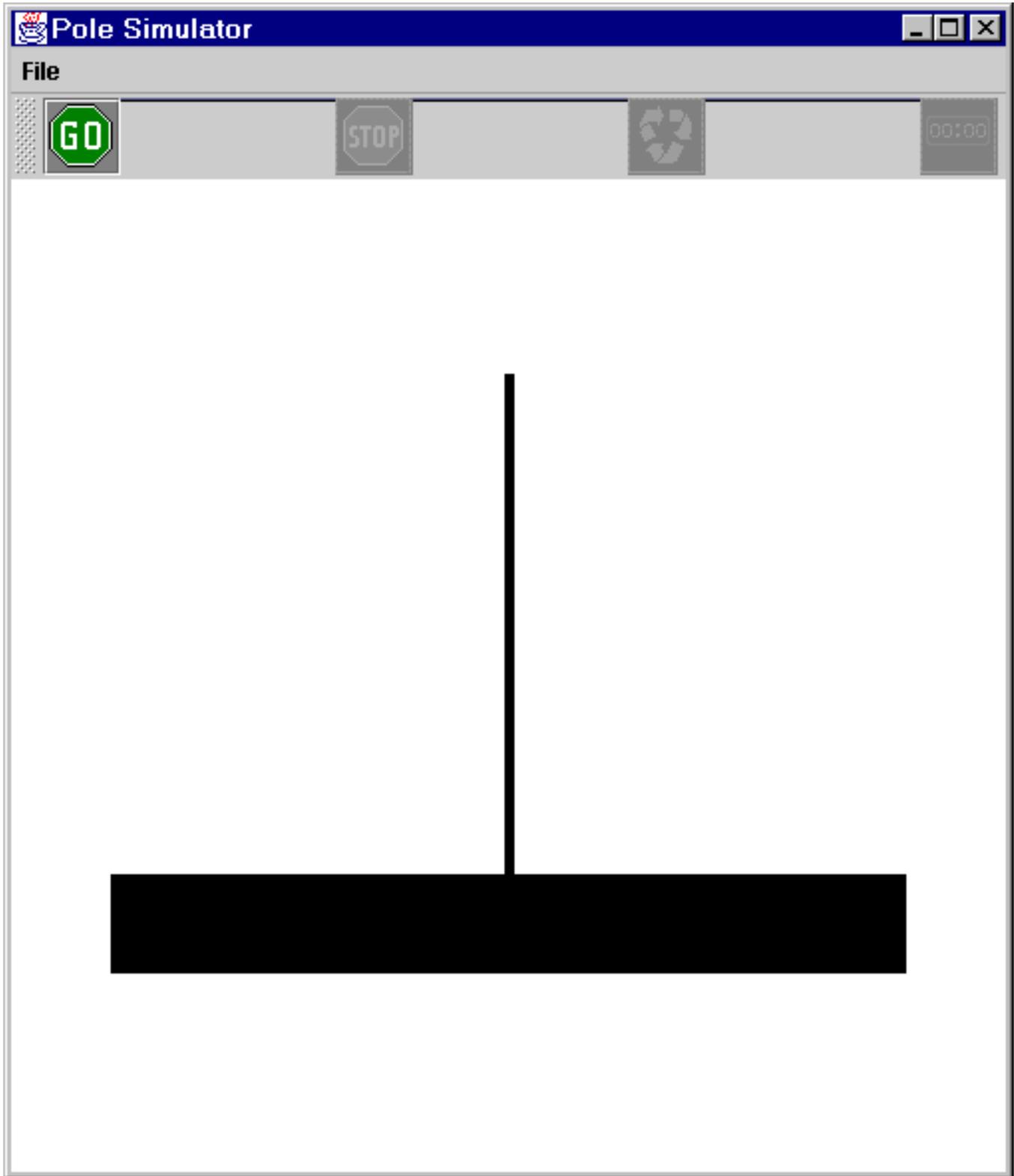


It should have a start (or go) button to start the simulation, a stop button to stop it, a continue button to continue a simulation that has been stopped and a reset button to start a simulation from the beginning, i. e. when time is equal to zero. All buttons should be disabled initially since no input data is available upon launching the program.



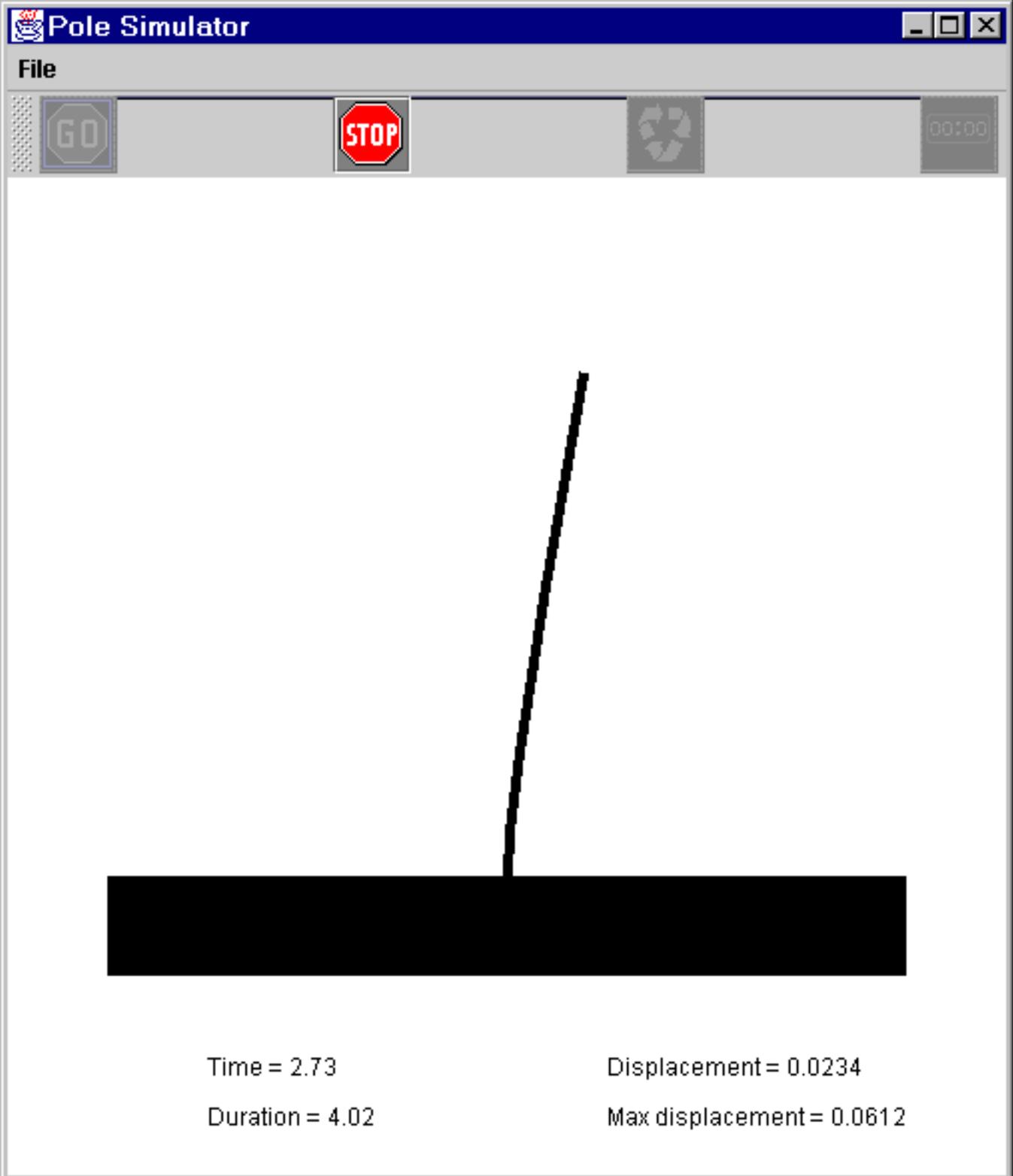
The program should have a menu bar with a menu named "File" with two menu items. The "Import Data" should allow the user to import, i.e. read, the data values provided in the above URL. The Exit should terminate the application and exit.

Having read the values from the file the "start", or "Go", button should be enabled to allow the user to start the simulation. The program should look as follows:



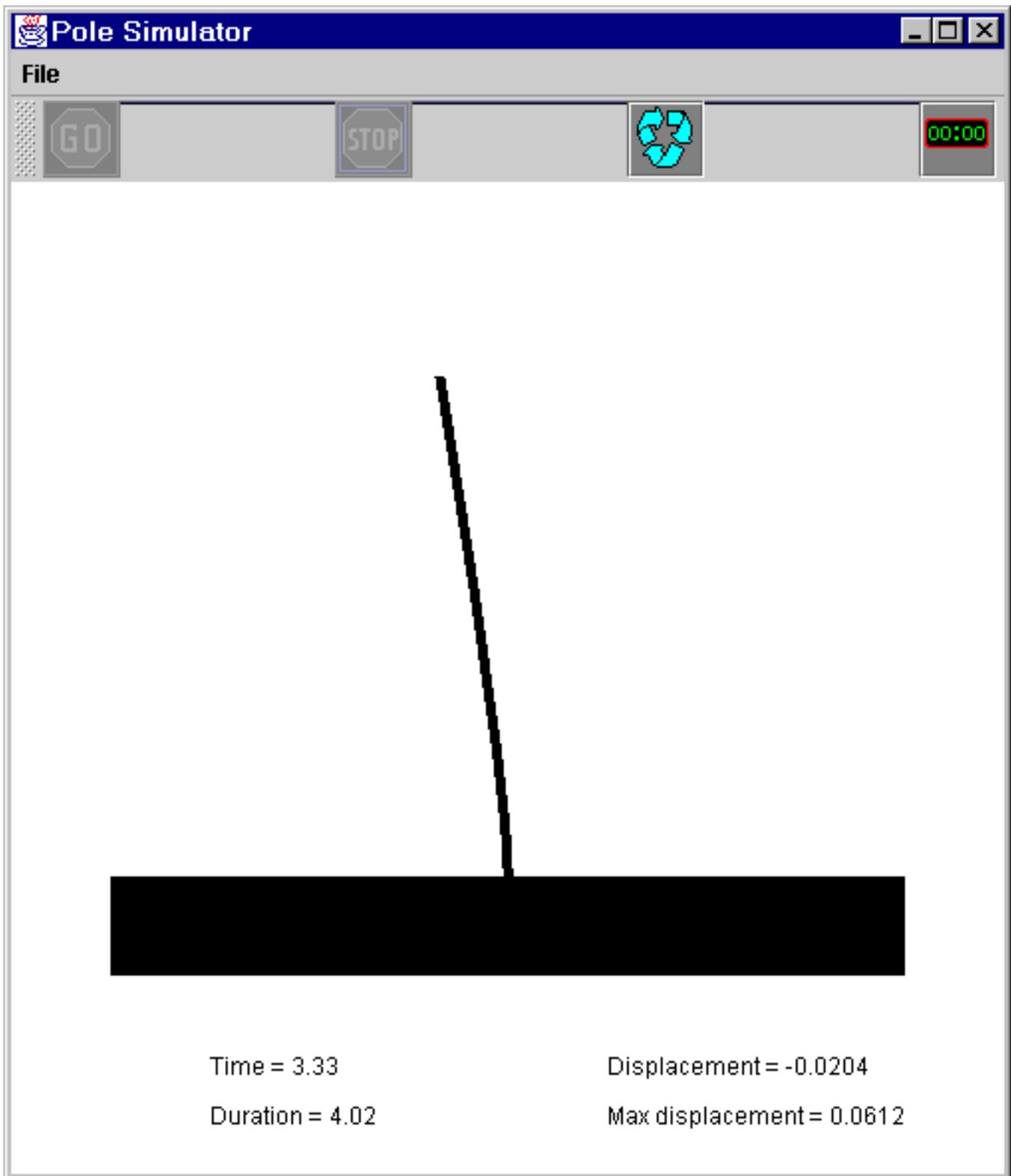
Upon pushing the start button the simulation should start. The pole should move according to the provided values of the displacements. In particular, the amplitude of movement should be such so that the maximum displacement in pixels be equal to 0.20 times the width of the panel that is used by the application to draw the pole. As soon as the simulation start the start button should be disabled and the stop button should be enabled to allow the user to stop the simulation at any time, as shown in the next snapshot of the simulation. Also, during simulation the time instance, the total duration of the

simulation, the displacement at the specific time instance and the maximum, in absolute value, displacement should be printed on the simulation panel, as shown below:

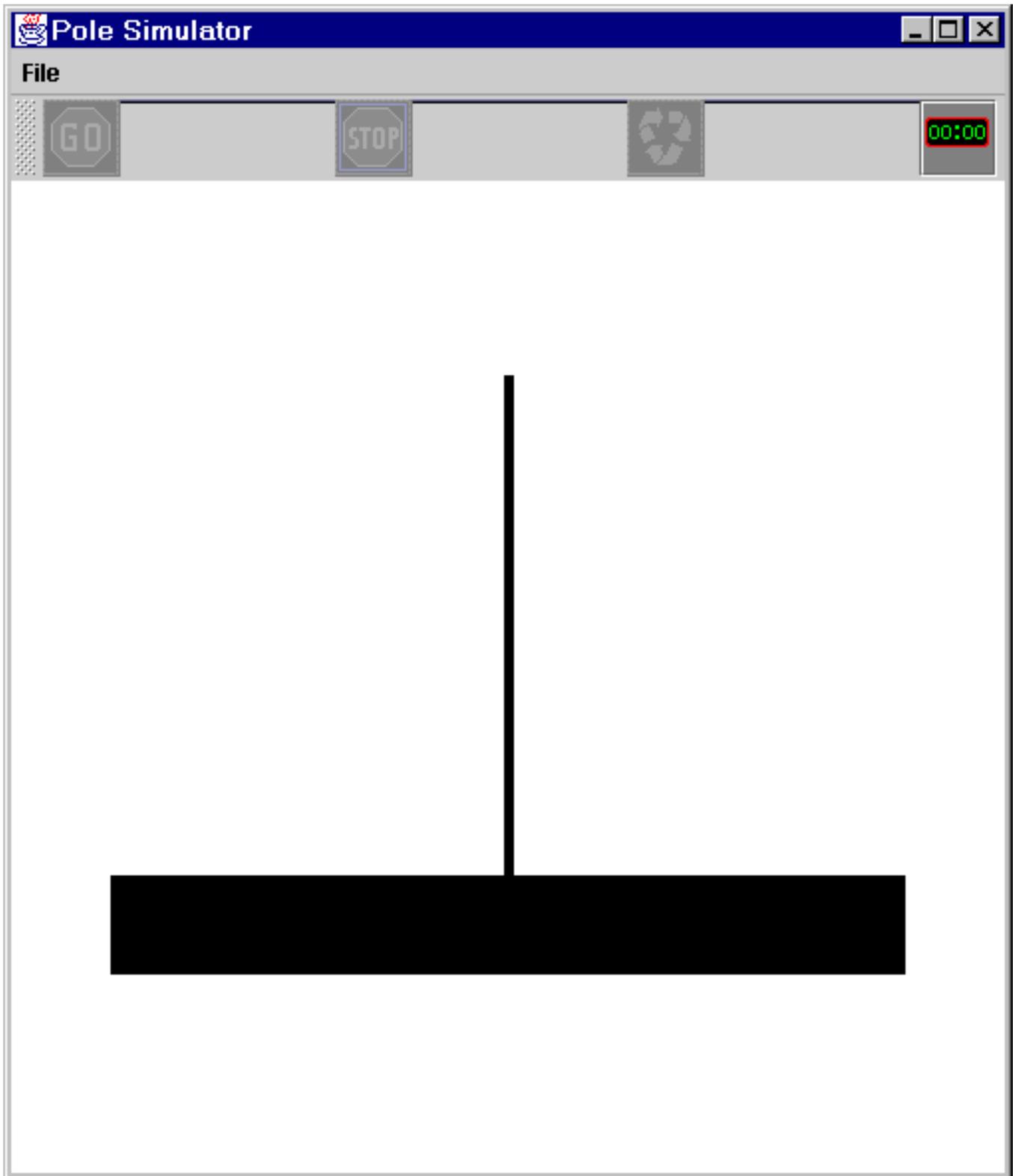


If the user clicks on stop to stop the simulation then the continue button should be enabled to allow the user to continue the simulation. Also the reset button should be enabled to allow the user reset the simulation. If the reset is selected then the simulation should start again from the beginning, i.e. from time zero. the following snapshot shows the continue and reset buttons being enabled, after the

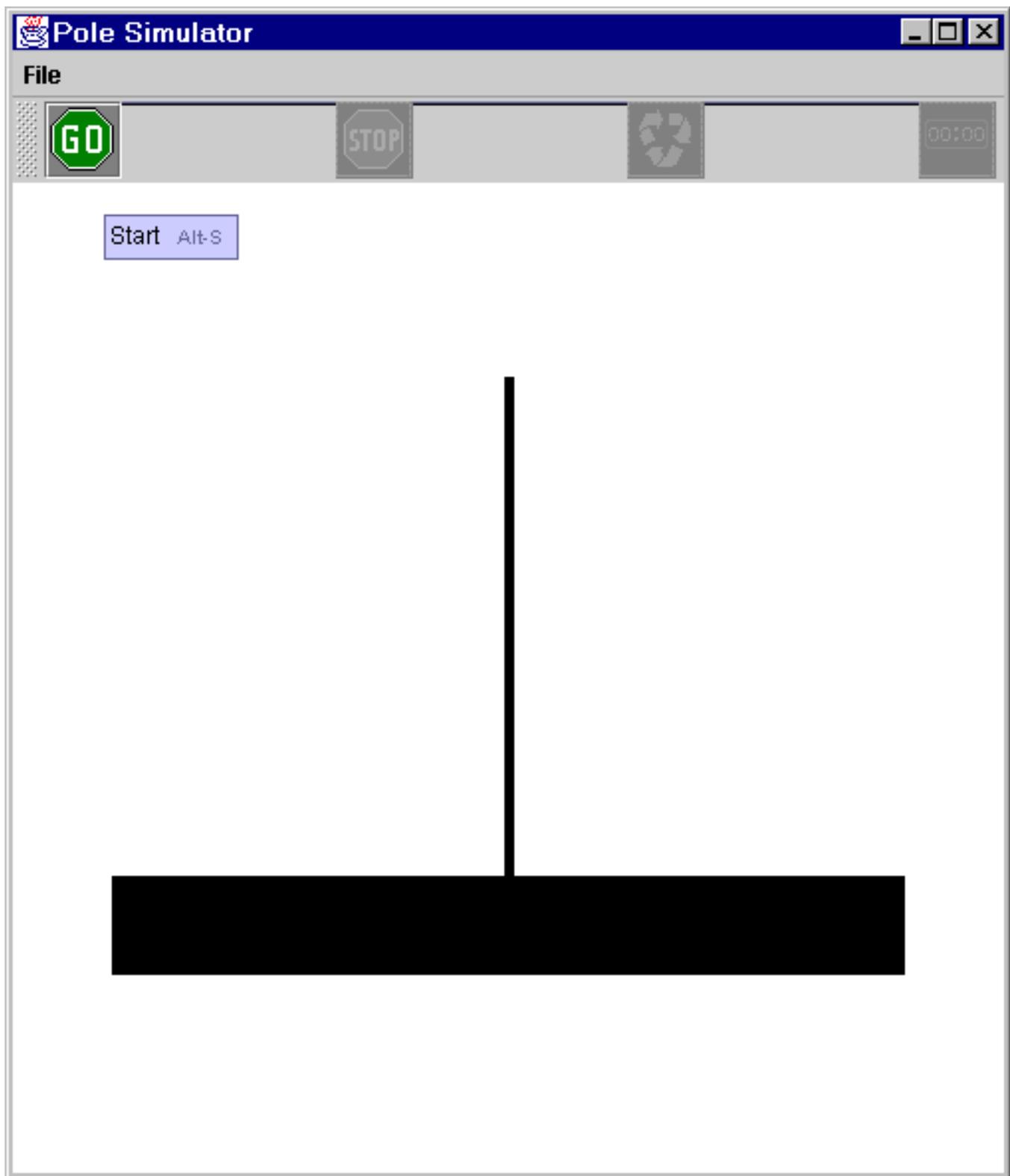
simulation has been stopped.



If the simulation is left to run until its end, i.e. until time reaches the duration of the input data then the reset button should be enabled to allow the resetting of the simulator, as shown below:



The start, stop, continue and reset buttons could also have shortcuts, i.e. specify mnemonic values to facilitate fast selection. You could also register text to be displayed in a tool tip as shown below.



You must submit, both in hardcopy form and electronically, only the Java source code files. You can use any number of files, as long as your program can be compiled with the command:

```
javac PoleSimulator.java
```

Your program should then be executed using the command:

Note:

Please submit **both** printouts of the source code you have written (preferably using `% enscript -2Gr -Pprinter filename`) and (or screen dumps of) the execution output (using `%xdpr -Pprinter`), with your name and username clearly written on the first page of the **stapled** submitted problem set. The submitted code must be identical to that electronically turned in (as described above).
