The Turtle System

Guide for Teachers

1. About this Guide

The *Guide for Normal Users* and the *Guide for Power Users* between them cover almost all of the functionality of the *Turtle System*. There are however a few features intended specifically for teachers, which we have separated out into this guide for convenience.

To gain access to these features, you first need to switch to Power User mode, by clicking on "Display Power User Menu" under the "View" menu. (For more information about Power User mode, see the *Guide for Power Users*.) The rest of this guide assumes that you have Power User mode enabled.

2. Make Marking Easier

With Power User mode enabled, three additional options become available in the "Options" menu: "Auto-Compile on loading", "Auto-Run on loading", and "Auto-Format on loading". By enabling these, and then saving the current options settings to your default options file (as explained in the *Guide for Normal Users*), you can potentially make marking your students' programs that much easier. For with these selected, when you open a Turtle program, the *System* will automatically attempt to compile it, showing you any errors if it fails, and otherwise format it in a standard way so that it is easier to read, and run it so that you can see its result—all without any extra button presses.

Furthermore, with Power User mode enabled you now have access to two source code tabs on the right hand side of the *System*: "Usage" and "Comments". The latter strips out and displays any comments in the program, while the former shows you a table of all the commands and command structures used in the program, helpfully categorised with totals and line numbers indicating where they appear. This way, if you set your students assignments with specific instructions such as, "use at least three relative movement commands", you can very quickly and easily check whether the program meets these requirements, freeing up your time for more constructive assessment and feedback.

3. Custom Examples

By default, the *Turtle System* comes with a large range of example programs, to illustrate the Turtle programming languages. In the classroom, however, some teachers may find that these are more of a distraction than a help. Alternatively, some may wish to modify or replace these examples with ones of their own. With the *Turtle System* version 11, this is now possible.

To begin with, under the "Options" menu (with Power User mode enabled), click the entry to "Save standard Help examples in file structure". This will export all of the built-in examples (as precompiled TGX files) into the directory containing the *Turtle System* executable file. Now click on the entry "Help menu to reflect file structure and examples". To all appearances, nothing will have changed. However, the *System* is now reading the "Help" menu from the examples saved to your computer, rather than from its built-in memory. If you browse to the directory containing the *Turtle System* executable file, you will see the example files in there.

Now you can edit, replace, or delete theses files to suit your needs. The changes will not take effect in the *System* immediately—to see them, you must either turn the option for the "Help menu to reflect file structure and examples" off and on again, or close and reopen the *Turtle System*. If doing the latter, remember to first save your current settings to the default options file.

The directory structure for the example programs should be self-explanatory. Every file and folder must be prefixed with an uppercase letter followed by an underscore (“A_”, “B_”, “C_”, etc.), and this is what fixes the order in which the categories and examples will appear in the menu. You can use ordinary program files (not precompiled TGX files) as your examples, but we recommend that you use the precompiled TGX format, since this also allows you to add metadata into your file (such as a name for the program). The *System* will display this name in the menu where available; otherwise it will display the filename.

4. Write Protection

To do any of the above, you must of course have write access in the directory where the *Turtle System* is installed. Should you wish to, however, you can set up the *System* with your own custom examples, and any other settings you want, save a default settings file, and then place everything in a directory on your school network to which students have read but not write access. That way, when students run the program from this directory, it will safely be set up the way that you want, and they will not be able to edit the example programs or the default settings themselves.

5. Disable Power User Mode

Having set up the *Turtle System* in the way that you want it—with your preferred options, and perhaps custom example programs—you may then, if you wish, disable Power User Mode *altogether*. This is different from simply switching Power User Mode off and returning to Normal User Mode. Rather, this switches Power User Mode off *and then hides the menu option to turn it back on again*. The point of this is to prevent your students from themselves enabling Power User Mode, and then for example reinstating the built-in example programs.

To disable Power User Mode altogether, click on “Disable Power User Options (interactively irreversible)” in the “View” menu. Make sure you have set up the *System* in the way that you want it first. It will now be impossible—using the *System* itself—to change any of the Power User options. In order to re-enable these options, you will need to edit (or delete) the “Default.tgo” options file in the directory containing the *Turtle System* executable. If you have placed this executable in a directory to which your students have read but not write access, this is something that they will be unable to do, and so you may be confident that the *System* is fixed in just the way that you want it.