

# ***Build A Better Mousetrap***

***By Amy Petré Hill***

Corporate software users are completely, unabashedly in love with the mouse. This palm-sized piece of plastic--with super-slick, roller-ball action and cute click buttons--instantly transforms even the greenest end user into a god. At every PC a deity now sits, gliding effortlessly across the GUI, cutting and pasting himself or herself into a frenzy, without the pitiful mortal encumbrances of keyboards or commands.

The mouse, and the colorful Windows universe it scuttles through, has created a computing Mount Olympus for end users. But to many software developers brought up on comfortable UNIX systems, the mouse can sometimes seem a pestilence from Hades, turning old-time coders into GUI "mouse catchers."

Developers comfortable with the Windows environment might find this analogy a little melodramatic, but for someone experiencing GUI for the first time, the environment can be overwhelming. The introduction of the mouse, and the graphical world in which it lives, has done more than just add additional clutter to desktops; it has demanded an entirely new approach to software development. Programs must now do more than respond to carriage returns, they must respond to a point or a click from a mouse on any part of the interface at any time. This new need has given rise to a whole new program structure, based on the *event loop*. The event loop sits at the heart of any GUI program and patiently waits for the user to do something, like click a button or a menu item. Once an action has been taken, the event loop instantly reacts.

Previously, creating event loops and working with graphical controls has been tedious. Even if a developer has heeded the subtle (and admittedly not always so subtle) urgings of *The BASIS Advantage* and looked into the theory behind event loops, there's a big difference between understanding the basic principles and transforming existing character-based programs into event-driven applications. Many developers have watched and waited for tools that can help them move their legacy programs into the brave new GUI world.

With Visual PRO/5® 2.0, BASIS delivers GUIBuilder™--the bridge between graphical interface theory and reality for which many developers have been waiting. This new tool can take graphical windows and controls created in ResBuilder™, associate them with event code, and then pull all the pieces together to create a complete program. GUIBuilder handles the housekeeping that goes into all event-driven GUI programs, leaving the developer with the more manageable task of incorporating business rules into an

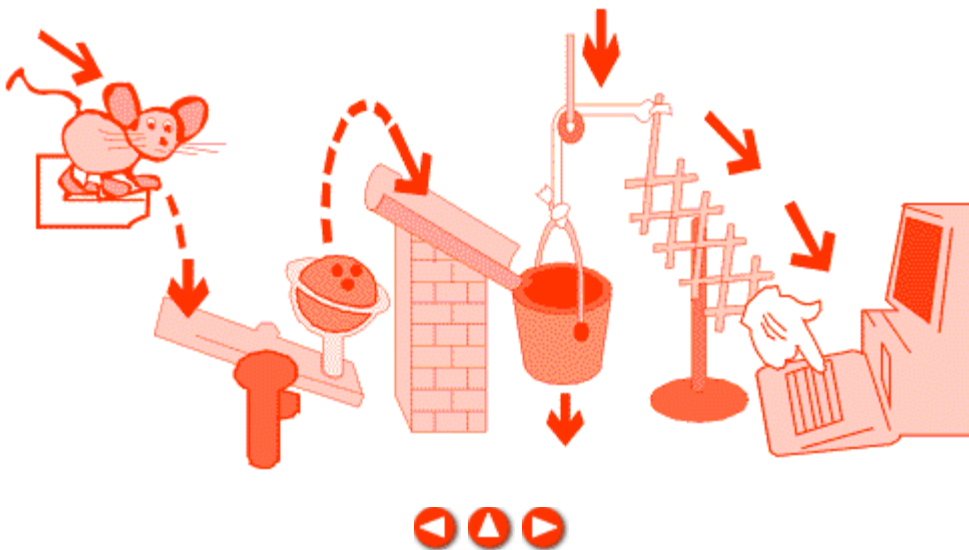
existing framework. Now developers have the tools they need to easily build that better mousetrap.

In this issue of *The BASIS Advantage*, Jim Douglas and William Baker, two of the key GUIBuilder developers, provide a comprehensive introduction to this new tool. Jim's articles, starting with "[GUIBuilder: The Easy Way to Go GUI](#)," offer an overview of GUIBuilder and its capabilities. In "[Randolph Grapples with GUIBuilder](#)," William chronicles Randolph's first experience with GUIBuilder while giving a step-by-step, introductory tour through this new tool.

In addition to in-depth coverage of mousetrap technique, Ernie Longmire discusses [strategies for exterminating that annoying Internet pest, spam](#), while Ed Holling offers another look inside Visual PRO/5 2.0 with an [overview of the new INPUTE and INPUTN verbs](#). And on a more international note, this issue welcomes Falk Spitzberg of BASIS International Software GmbH as a new *Advantage* writer. Falk provides an interesting [personal account of the recent CeBit show in Germany and the creation of the new BASIS subsidiary](#).

As always, we would like to hear how we're doing, so please send us your comments and suggestions. We also encourage you to check out the new [online edition of The BASIS Advantage](#) at [www.basis.com/advantage/index.html](http://www.basis.com/advantage/index.html).

Enjoy!



[Copyright 1999, BASIS International Ltd. All rights reserved.](#)  
[Terms of Use.](#)