

Playing Minesweeper automatically with Perl

by Paul J. Fenwick
pjf@perltraining.com.au



You sit down at the machine, it's running Windows. Those other operating systems are pretty good, but they're lacking that one killer app, the one reason why Windows still has market dominance: Minesweeper!

Yes, good old Minesweeper! You'll just play one game, on intermediate and then start on your real work. Just a little morsel of relaxation. Three hours later, you're wondering what happened to your day.

Does this story sound familiar or have you a friend or work-mate who's in the jaws of the Minesweeper vice? I know that I was, so one day, I decided to do something about it. I'd program my machine to play Minesweeper for me.

I'd automated the rest of my life; I have programs to read my mail, to plan my exercise, and to pay my bills. Automating my leisure time was the obvious next step. Having a program to play Minesweeper for me would give me more time for the more important things in life, like Solitaire. It was from this realisation that App: :SweeperBot was born.

■ Why a Minesweeper bot? -----

I'm a lazy person. I hate having to do things myself, and I really hate having to write my own code. Indeed, one of the motivating factors for writing App: :SweeperBot is that much of the hard work had already been done for me.

Matt Sparks (<http://www.forked.com>) had already written an almost complete Minesweeper bot, doing what I had considered all the "hard parts". Matt's code was able to locate the board, start a new game, and identify the contents of various squares. All I had to do was to write an algorithm to get it to play.

Of course, when asked by serious professionals with lots of money why I wrote App: :SweeperBot, I claim it's because it provides an excellent working example of being able to examine and control GUI applications using Perl. These techniques can be a great advantage in testing and automation.

■ Locating Minesweeper -----

The first step for SweeperBot in playing Minesweeper is to locate the game on the screen. SweeperBot uses the Win32::GuiTest module for GUI manipulation, and this also makes locating the Minesweeper board easy:

```
use Win32::GuiTest qw(FindWindowLike);

my ($window_id) = FindWindowLike(
    0, '^Minesweeper' );
```



FindWindowLike returns a list of matching windows, but I'm only interested in the first one. The first argument to FindWindowLike is the root of my window search; it can be an existing window if I'm looking for a dialog box or similar window spawned by an application, but for SweeperBot we're using 0 to look for any window that matches my criteria.

The second argument is a regexp that matches my window name. In the case of SweeperBot, I'm simply looking for any window that starts with Minesweeper.

Once I've found my window, the next task is to determine how large it is; SweeperBot needs this information to determine the size of the Minesweeper grid, and to locate the all-important smiley face that's used to determine our game state (won/lost/playing) and to reset the game.

```
use Win32::GuiTest qw(GetWindowRect);

my ($left, $top, $right, $bottom)
    = GetWindowRect($window_id);

my $width  = $right - $left;
my $height = $bottom - $top;
```