Subject: unit testing in U++
Posted by exolon on Fri, 27 Oct 2006 02:16:17 GMT
View Forum Message <> Reply to Message

Luzr mentioned that unit testing might be a future addition to U++'s arsenal.

So I guess this is as good a place as any to discuss it, having not found anything elsewhere on the forums.

A good introductory read on the state of C++ unit testing (albeit 2 years ago) is here - that guy went on to write his own testing framework, UnitTest++, since he uses test-driven design in his job. I was thrown into TDD in Java while on a work placement this Spring/Summer, and want to embrace it as much as possible in C++ with U++ if I can.

I'm not sure which approach is best, since C++ is an entirely different beast to Java. A big weakness with most testing frameworks in C++ IMO is that they don't allow mock object testing, which was truly useful in our work. Apart from that, there are a lot of good approaches...

Input welcome!

Subject: Re: unit testing in U++
Posted by mirek on Sun, 29 Oct 2006 03:51:07 GMT
View Forum Message <> Reply to Message

Well, I did not knew much about "test frameworks" before and I do know a little now.

Anyway, it seems to me that quite interesting option would be to employ TheIDE and package system to do tests, while perhaps storing test results in Sqlite database or something like that.

In other words, perhaps the TestCase would be a package, TestSuite a nest and all results stored to be visible by TheIDE. TheIDE then would be able to compile and run all package in testsuite nest and collect results.

Mirek

Subject: Re: unit testing in U++
Posted by fallingdutch on Sun, 29 Oct 2006 06:23:38 GMT
View Forum Message <> Reply to Message

Sounds very good, Mirek, but i don't know wether it is possible, because you need to write Testfunctions for all of your methods and new classes you add, so maybee with a subdirectory ... and the functions/packages that will be checked with unit test should be selecteable .... I have eg done test with 10000 loop with random data ... would take a lot of time to check every function of

U++ with testcases like that

Bas

Subject: Re: unit testing in U++
Posted by mirek on Sun, 29 Oct 2006 17:00:13 GMT
View Forum Message <> Reply to Message

fallingdutch wrote on Sun, 29 October 2006 01:23Sounds very good, Mirek, but i don't know wether it is possible, because you need to write Testfunctions for all of your methods and new classes you add, so maybee with a subdirectory ... and the functions/packages that will be checked with unit test should be selecteable .... I have eg done test with 10000 loop with random data ... would take a lot of time to check every function of U++ with testcases like that

Bas

Actually, we are not exactly speaking about unit testing of U++, but unit test framework within U++...

BTW, I still have my doubts about usefulness of unit testing for GUI classes... (how are you about to detect rare visual glitches?

Mirek

Subject: Re: unit testing in U++
Posted by exolon on Mon, 30 Oct 2006 01:20:00 GMT
View Forum Message <> Reply to Message

luzr wrote on Sun, 29 October 2006 17:00
BTW, I still have my doubts about usefulness of unit testing for GUI classes... (how are you about to detect rare visual glitches?
Yeah, that's a difficult case. In general, it's much easier to apply it to more logical stuff, although it's handy for test-driven design if you can mock certain calls.

About the packaging and stuff, I'm not sure. So far, I've been creating a separate project (ie: Palindrome and PalindromeTest) because I wasn't sure how to nicely separate test and tested code.. it's easier with java's package naming system and multiple source directories.
Also, running the tests means compiling all the tests into a separate executable and running that. I'm not sure how to manage that stuff properly yet...

Something like the junit test runners in Eclipse, for example, would be amazing (you select a test (or many) and hit alt-shift-X, t, and it runs the test methods, showing you a progress bar that turns red for errors or green for passes... quick and nice!).

Subject: Re: unit testing in U++
Posted by mirek on Mon, 30 Oct 2006 06:11:13 GMT

View Forum Message <> Reply to Message

exolon wrote on Sun, 29 October 2006 20:20
Something like the junit test runners in Eclipse, for example, would be amazing (you select a test (or many) and hit alt-shift-X, t, and it runs the test methods, showing you a progress bar that turns red for errors or green for passes... quick and nice!).

Well, I was even thinking about something like overnight tests (test server downloads sources from versioning system, runs tests, sends report by email).

BTW, it is my impression that in fact, errors must be quite rare while testing...

Mirek

Subject: Re: unit testing in U++
Posted by exolon on Tue, 31 Oct 2006 05:47:14 GMT

View Forum Message <> Reply to Message

luzr wrote on Mon, 30 October 2006 06:11Well, I was even thinking about something like overnight tests (test server downloads sources from versioning system, runs tests, sends report by email).
Well, sure. You might configure CVS to cause a build and test for every check in. Working over the summer, we had CruiseControl run continuous builds of all projects on the build server, automatically... when you checked in changes that broke anything, it'd email us, or for some of us (until we turned off notifications) send SMS to our phones

luzr wrote on Mon, 30 October 2006 06:11BTW, it is my impression that in fact, errors must be quite rare while testing...
Depends on how you code and test. With test-driven design, the way I like to code, you write a little bit of test code BEFORE that functionality exists, then run that test, confirm that it fails... then write the smallest piece of code to pass that test. Then you watch it pass, and repeat the process until you've implemented what you wanted. Usually works well for me, but some things are harder to test, of course.

About some of the GUI stuff (and more), there is some discussion of it starting here.

Subject: Re: unit testing in U++
Posted by ebojd on Mon, 05 Feb 2007 16:21:46 GMT

View Forum Message <> Reply to Message

A number of points...

There are actually a couple of software packages available which will do automated GUI and command line interface testing using something akin to an expect/send concept. If you google around you would likely find a couple that may be useful. I feel I must bow out of discussing this particular subject as I have worked for one of the companies. And even though they canceled my position in the middle of my relocation trip after serving my probationary period (Oh yes, and they invited me to reapply for the same position), I still respect the non disclosure...

The writing tests before code that exolon is referring to is part of a programming paradigm called eXtreme Programming (XP) <see http://www.extremeprogramming.org/ for details, specifically the section on writing the test code first http://www.extremeprogramming.org/rules/testfirst.html>

Subject: Re: unit testing in U++
Posted by exolon on Mon, 05 Feb 2007 21:47:40 GMT
View Forum Message <> Reply to Message

ebojd wrote on Mon, 05 February 2007 16:21
And even though they canceled my position in the middle of my relocation trip after serving my probationary period (Oh yes, and they invited me to reapply for the same position)
Yowch!

Subject: Re: unit testing in U++
Posted by ebojd on Mon, 05 Feb 2007 23:15:38 GMT
View Forum Message <> Reply to Message

yep... in addition to having half my stuff already located over 2500 miles away (oh I forgot to add at my own expense), it cost me some things personally that cannot be judged monetarily. In the end not only did I not reapply for the post but I refused to come back even as a contract consultant. It was a sore point, but I still respect the confidentiality.

Well, enough of this off topic thread...   Here's to wishing you all never experience the likes of that day!