Subject: Squirrel - the programming language
Posted by Sender Ghost on Thu, 24 Nov 2011 23:31:27 GMT
View Forum Message <> Reply to Message

Homepage:
http://squirrel-lang.org

License:
MIT

Version:
3.0.7 (stable)

Description:
Squirrel is a high level imperative, object-oriented programming language, designed to be a light-weight scripting language that fits in the size, memory bandwidth, and real-time requirements of applications like games.
Although Squirrel offers a wide range of features like dynamic typing, delegation, higher order functions, generators, tail recursion, exception handling, automatic memory management, both compiler and virtual machine fit together in about 6k lines of C++ code.

Documentation:
Squirrel 3.0 reference manual
Squirrel 3.0 Standard Libraries manual

In the attachments you could find Squirrel source code, samples, converted to U++ packages and documentation.

To note:
There were minor changes for include files for original sources to adapt for U++ package structure.
The "squirrel.h" file includes "config.h" file with double precision define (SQUSEDOUBLE) by default.

Edit: Updated to 3.0.7 version.

File Attachments
1) squirrel_v3.0.7.zip, downloaded 396 times

Subject: Sqrat - Squirrel C++ Binding Utility
Posted by Sender Ghost on Thu, 24 Nov 2011 23:33:42 GMT
View Forum Message <> Reply to Message

Homepage:
http://scrat.sourceforge.net

License:

zlib/libpng

Version:
0.9.1

Description:
Sqrat is a C++ library for Squirrel that facilitates exposing classes and other native functionality to Squirrel scripts. It is similar to SqPlus, both in functionality and syntax, but seeks to address several issues present in other binding libraries.

Sqrat models the underlying Squirrel API a little more closely than most other binding utilities, and this fact gives it a lot of power in exchange for a couple of minor quirks in the syntax. Code is straightforward and easy to read, however, and can easily be ported to from existing bindings.

Beginning in Sqrat 0.8, several utilitiy libraries are also included to extend the squirrel language itself with useful libraries and functionalities.

Sqrat has been tested against Squirrel 2.1.1 through 2.2.3 and 3.0 to 3.0.4. Other versions may work fine, but the author makes no promises!

Documentation:
Binding Library
Module Import Library
Threading Library
Doxygen generated documentation for 0.9 version

In the attachments you could find Sqrat source code, sq sample, converted to U++ packages.

To note:
There were minor changes for include files for original sources to adapt for U++ package structure and disabling some of MSC warning about comments inside of function arguments.

Edit: Updated to 0.9.1 version.

File Attachments

1) sqrat_v0.9.1.zip, downloaded 399 times

Subject: Re: Squirrel - the programming language
Posted by Sender Ghost on Thu, 24 Nov 2011 23:35:59 GMT
View Forum Message <> Reply to Message

As for AngelScript, I made sample application based on Mirek's "little experiment" example.

With following results:
1 / (1 - x * y + x - y) = -0.01851851852
fn->Execute() = -0.01851851852
sum = 5190404.858

sum = 5190404.858
sum = 5190404.858
sum = 5190404.858
TIMING Squirrel (fully interpreted): 197.00 ms - 197.00 ms (197.00 ms / 1), min: 197.00 ms, max: 197.00 ms, nesting: 1 - 1
TIMING Direct : 13.00 ms - 13.00 ms (13.00 ms / 1), min: 13.00 ms, max: 13.00 ms, nesting: 1 - 1
TIMING Compiled : 57.00 ms - 57.00 ms (57.00 ms / 1), min: 57.00 ms, max: 57.00 ms, nesting: 1 - 1
TIMING Interpreted : 770.00 ms - 770.00 ms (770.00 ms / 1), min: 770.00 ms, max: 770.00 ms, nesting: 1 - 1

Edit: Updated to Squirrel 3.0.7 version. The results from previous versions.

File Attachments

1) UppCompiler_with_Sqrat.zip, downloaded 371 times

Subject: Re: Squirrel - the programming language
Posted by mdelfede on Fri, 25 Nov 2011 06:43:00 GMT
View Forum Message <> Reply to Message

Hi Sender Ghost,

thank you for posting the packages

BTW, a question.... I still don't understand the timing with RTIMING macro... could you explain a bit more the results ?

For example, what are the 197 ms for fully interpreted and the 770 ms for interpreted ?

Max

Subject: Re: Squirrel - the programming language
Posted by Sender Ghost on Fri, 25 Nov 2011 06:58:55 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Fri, 25 November 2011 07:43
BTW, a question.... I still don't understand the timing with RTIMING macro... could you explain a bit more the results ?

For example, what are the 197 ms for fully interpreted and the 770 ms for interpreted ?

Hello, Massimo.

RTIMING is profiling macro for source code scopes. In this case for:

```
{
 RTIMING("Interpreted");
 double sum = 0;
 // ...
 RDUMP(sum);
}
// ...
{
 RTIMING("Squirrel (fully interpreted)
 using namespace Sqrat;

 double sum = 0;
 // ...
 RDUMP(sum);
}
```

RTIMING writes results to application *.log file, even in Release mode. TIMING for Debug.

Subject: Re: Sqrat - Squirrel C++ Binding Utility
Posted by unodgs on Fri, 25 Nov 2011 07:10:43 GMT
View Forum Message <> Reply to Message

That binding utility looks interesting. I think I'll give squirrel a try

---

Subject: Re: Squirrel - the programming language
Posted by mdelfede on Fri, 25 Nov 2011 07:42:58 GMT
View Forum Message <> Reply to Message

Sender Ghost wrote on Fri, 25 November 2011 07:58mdelfede wrote on Fri, 25 November 2011 07:43
BTW, a question.... I still don't understand the timing with RTIMING macro... could you explain a bit more the results ?

For example, what are the 197 ms for fully interpreted and the 770 ms for interpreted ?

Hello, Massimo.

RTIMING is profiling macro for source code scopes. In this case for:

```
{
 RTIMING("Interpreted");
 double sum = 0;
 // ...
```

```
 RDUMP(sum);
}
// ...
{
 RTIMING("Squirrel (fully interpreted)
 using namespace Sqrat;

 double sum = 0;
 // ...
 RDUMP(sum);
}
```

RTIMING writes results to application *.log file, even in Release mode. TIMING for Debug.

The time given is just the time spent inside block ? Because I see than that Squirrel fully interpreted is quite faster than interpreted case.....
BTW, looking into your example, it seems to me that the squirrel time is the whole of compiling+running the script (why running ?) and evaluating the function; it would be more interesting to take evaluation part of pre-compiled script separate.

I mean, usually script compiling is done once and the important stuff is execution speed.

@Unodgs : I like squirrel because it's simple, powerful and it forces you to declare variables, even if untyped.... I hate javascript because of this lack which makes very easy to introduce hidden bugs just because of mispelled vars.

Max

Subject: Re: Squirrel - the programming language
Posted by Sender Ghost on Fri, 25 Nov 2011 07:55:07 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Fri, 25 November 2011 08:42
BTW, looking into your example, it seems to me that the squirrel time is the whole of compiling+running the script (why running ?) and evaluating the function; it would be more interesting to take evaluation part of pre-compiled script separate.

Because without running, it will not find Squirrel function to get interpreted result.
Results for pre-compiled code:

```
1 / (1 - x * y + x - y) = -0.01851851852
fn->Execute() = -0.01851851852
sum = 5190404.858
sum = 5190404.858
sum = 5190404.858
sum = 5190404.858
```

TIMING Squirrel (fully interpreted): 195.00 ms - 195.00 ms (195.00 ms / 1), min: 195.00 ms, max: 195.00 ms, nesting: 1 - 1
TIMING Direct : 13.00 ms - 13.00 ms (13.00 ms / 1), min: 13.00 ms, max: 13.00 ms, nesting: 1 - 1
TIMING Compiled : 58.00 ms - 58.00 ms (58.00 ms / 1), min: 58.00 ms, max: 58.00 ms, nesting: 1 - 1
TIMING Interpreted : 759.00 ms - 759.00 ms (759.00 ms / 1), min: 759.00 ms, max: 759.00 ms, nesting: 1 - 1

Results by sections:
Toggle Spoiler

```
 using namespace Sqrat;

 double sum = 0;
 const String text =
 "function compute() {\n"
 " local x, y, sum = 0.0;\n"
 " for (x = 0.0; x < 1; x += 0.001)\n"
 "  for (y = 0.0; y < 1; y += 0.001)\n"
 "   sum += 1 / (1 - x * y + x - y);\n"
 " return sum;\n"
 "}\n";

 // creates a VM with initial stack size 1024
 HSQUIRRELVM vm = sq_open(1024);

 DefaultVM::Set(vm);

 Script script;
 {
  RTIMING("Squirrel (compiling)")
  try {
   script.CompileString(~text);
  }
  catch (Error ex) {
   Cerr() << "Exception: " << ex.Message(vm).c_str() << '\n';
   SetExitCode(1);
   return;
  }
 }

 Function compute;

 {
  RTIMING("Squirrel (running and getting function)")
  try {
   script.Run();
```

```
 }
 catch(Error ex) {
  Cerr() << "Exception: " << ex.Message(vm).c_str() << '\n';
  SetExitCode(1);
  return;
 }

 compute = RootTable().GetFunction("compute");
}

{
 RTIMING("Squirrel (fully interpreted)")
 if (!compute.IsNull()) {
  try {
   SharedPtr<double> val = compute.Evaluate<double>();
   sum = *val;
  }
  catch (Error ex) {
   Cout() << "Exception: " << ex.Message(vm).c_str() << '\n';
   SetExitCode(1);
   return;
  }
 }
}

 //sq_close(vm);
 RDUMP(sum);
```

```
1 / (1 - x * y + x - y) = -0.01851851852
fn->Execute() = -0.01851851852
sum = 5190404.858
sum = 5190404.858
sum = 5190404.858
sum = 5190404.858
TIMING Squirrel (fully interpreted): 222.00 ms - 222.00 ms (222.00 ms / 1 ), min: 222.00 ms, max:
222.00 ms, nesting: 1 - 1
TIMING Squirrel (running and getting function):  0.00 ns -  0.00 ns ( 0.00 ns / 1 ), min:  0.00 ns,
max:  0.00 ns, nesting: 1 - 1
TIMING Squirrel (compiling):  0.00 ns -  0.00 ns ( 0.00 ns / 1 ), min:  0.00 ns, max:  0.00 ns,
nesting: 1 - 1
TIMING Direct         : 14.00 ms - 14.00 ms (14.00 ms / 1 ), min: 14.00 ms, max: 14.00 ms, nesting:
1 - 1
TIMING Compiled       : 59.00 ms - 59.00 ms (59.00 ms / 1 ), min: 59.00 ms, max: 59.00 ms,
nesting: 1 - 1
TIMING Interpreted    : 848.00 ms - 848.00 ms (848.00 ms / 1 ), min: 848.00 ms, max: 848.00 ms,
```

nesting: 1 - 1

Edit: Updated to Squirrel 3.0.7 version. The results from previous versions.

File Attachments
1) UppCompiler_with_Sqrat_2.zip, downloaded 344 times

Subject: Re: Squirrel - the programming language
Posted by mdelfede on Fri, 25 Nov 2011 08:21:24 GMT
View Forum Message <> Reply to Message

Uhmmm... it seems that the compiling time is around zero... weird.
Anyways, it shows that running a pre-compiled script don't bring advantages, at least for such simple scripts.

BTW, it looks slower than Angelscript, but by far easier to use.

Max

Subject: Re: Squirrel - the programming language
Posted by Sender Ghost on Fri, 25 Nov 2011 08:35:46 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Fri, 25 November 2011 09:21
but by far easier to use.

Not quite true.
The AngelScript syntax is near with C++. It is possible to create wrapper functions to get AngelScript functions and use assert(r >= 0); for result checks. What I did is just complete code. This is because I added Sqrat binding library. Using Squirrel stacks directly will show different picture.

They are just different.

Subject: Re: Squirrel - the programming language
Posted by mdelfede on Fri, 25 Nov 2011 08:48:00 GMT
View Forum Message <> Reply to Message

Yep, of course, I meant squirrel + sqrat, not squirrel alone.

Max

Subject: Re: Squirrel - the programming language
Posted by unodgs on Fri, 25 Nov 2011 13:24:00 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Fri, 25 November 2011 02:42
@Unodgs : I like squirrel because it's simple, powerful and it forces you to declare variables, even if untyped.... I hate javascript because of this lack which makes very easy to introduce hidden bugs just because of mispelled vars.

I tried this binding utility and I must say I didn't expect it to be so easy.

```
struct Test {
    void SayHello() {
        Cout() << "Hello\n";
    }
};

.. bindings ...

RootTable(vm).Bind("Test", Class<Test>(vm)
    .Func("SayHello", &Test::SayHello)
);

Test globTest;
RootTable(vm).SetInstance("globTest", &globTest);


.. and the script ..

local test = Test();
test.SayHello();
globTest.SayHello();
```

This is just awesome! As for JS - I'm not a huge fan either (I don't like broken == operator and few other weird things) but if it comes to declaring variables you can use var everywhere and new strict mode. That should help.

@SenderGhost: Thank you for sharing Squirrel language - I didn't know this one.