Subject: [Proposal] The maintainer and the future of TURTLE package
Posted by Oblivion on Tue, 14 Jul 2020 16:55:10 GMT
View Forum Message <> Reply to Message

Hi,

I was wondering if TURTLE is actively maintained. I value this package, and I find it useful for my purposes.

Does it have a maintainer? IF not, I'd be happy to step up to be its maintainer.

My initial plan is to move it to VirtualGUi.

I already have an experimental version of it based it on VirtualGui.

However, VirtualGui seems to define DoPaint and SyncCaret methods, And they are not overrideable. Is there any other way, in which I can ge text cursor (caret) and mouse cursor information to send over the wire protocol?

Best regards,
Oblivion

Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by mirek on Tue, 28 Jul 2020 15:42:47 GMT
View Forum Message <> Reply to Message

Excellent! You should now have write rights in the svn...

Mirek

Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by Oblivion on Wed, 29 Jul 2020 09:00:52 GMT
View Forum Message <> Reply to Message

Thank you!

My question remains though:

 VirtualGui seems to define DoPaint and SyncCaret methods, And they are not overrideable. Is there any other way, in which I can ge text cursor (caret) and mouse cursor information to send over the wire protocol?

This can be worked around by preprocessor commands (for example), but that would be ugly, and

an inelegant way to do so.

Best regards,
Oblivion

Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by mirek on Wed, 29 Jul 2020 09:15:52 GMT
View Forum Message <> Reply to Message

Well, there seems to be SystemCursor bool variable that makes VirtualGui use SetMouseCursor
method. I think we can have SystemCaret / SetCaret as well. That would solve the problem,
correct?

Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by Oblivion on Thu, 30 Jul 2020 18:51:19 GMT
View Forum Message <> Reply to Message

Hello Mirek,

Quote:. I think we can have SystemCaret / SetCaret as well. That would solve the problem,
correct?

Yeah, I think this should be the way to send the required caret information.
SetCaret method along with a static SystemCaret boolean would be nice to have.

Best regards,
Oblivion

Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by mirek on Fri, 31 Jul 2020 08:37:11 GMT
View Forum Message <> Reply to Message

Implemented (not quite tested SetCaret, but should work), I have also replaced SystemCursor
with GetOptions method....

Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by Oblivion on Fri, 31 Jul 2020 14:05:43 GMT
View Forum Message <> Reply to Message

OK, thanks!

Hopefully I'll be able to finish the port to VirtualGui next week.

Best regards,
Oblivion

Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by Oblivion on Wed, 26 Aug 2020 15:12:49 GMT
View Forum Message <> Reply to Message

Hello Mirek,

I have ancountered a curious problem while testing the Turtle/VirtualGui port.
(Note: This is tested on a Linux machine with default LANG=UTF-8 encoding. and this problem affects both old and new Turtle packages as the VirtualGui package shares the same clipboard code.)

The problem is GetWString and GetString do not work properly if called on a clipboard text.

TextCtrl derived classes are affected by this, as they use the same mechanism.

Sreenshot with explanation:

Now, the culprit seems to be the VirtualGui clipboard String/WString conversion code (VirtualGui/Clip.cpp, 64-86):

```
String GetString(PasteClip& clip)
{
 GuiLock __;
 if(clip.Accept("wtext")) {
  String s = ~clip;
  return WString((const wchar *)~s, wstrlen((const wchar *)~s)).ToString(); / << Adds random
chars. Why is this though? It is already a string. For charset conversion?

        //  return ToUtf16(s).ToString();    // Seems to work fine...
        //  return s.ToWString().ToString(); // Seems to work fine...

 }
 if(clip.IsAvailable("text"))
  return ~clip;
 return Null;
}

WString GetWString(PasteClip& clip)
{
 GuiLock __;
```

```
if(clip.Accept("wtext")) {
 String s = ~clip;
 return WString((const wchar *)~s, wstrlen((const wchar *)~s)); // Adds random chars...

        //  return ToUtf16(s);    // Seems to work fine...
        //  return s.ToWString(); // Seems to work fine...

 }
 if(clip.IsAvailable("text"))
  return (~clip).ToWString();
 return Null;
}
```

Best regards,
Oblivion

File Attachments
1) Turtle-paste-error.png, downloaded 691 times

Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by mirek on Thu, 27 Aug 2020 08:35:21 GMT
View Forum Message <> Reply to Message

The code is copied from Win32 where there can be some data after double zero terminator. I think problem here is that there is just single-zero terminator...

I guess your fix is fine.

Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by Oblivion on Fri, 28 Aug 2020 12:59:05 GMT
View Forum Message <> Reply to Message

Hello Mirek,

Thank you for your help and patience.

Before I finalize the Turtle port, I need a clarification on one point:

The original Turtle package uses "all-static" approach.

1) Is it OK to keep the VirtualGui-based Turtle all-static too (It works well...), or should I remove

the static stuff and make them instance members?
2) Accordingly, can I instantiate more than one VirtualGui/TurtleGui in the same process. Does U++ properly support this?
   if so, this might prove useful for later iterations of the Turtle package.

Best regards,
Oblivion

Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by mirek on Fri, 28 Aug 2020 13:41:49 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Fri, 28 August 2020 14:59Hello Mirek,

Thank you for your help and patience.

Before I finalize the Turtle port, I need a clarification on one point:

The original Turtle package uses "all-static" approach.

1) Is it OK to keep the VirtualGui-based Turtle all-static too (It works well...), or should I remove the static stuff and make them instance members?
2) Accordingly, can I instantiate more than one VirtualGui/TurtleGui in the same process. Does U++ properly support this?
   if so, this might prove useful for later iterations of the Turtle package.

Best regards,
Oblivion

Well, it is still using CtrlCore and CtrlCore has a lot of global variables, so it does not really make sense to bother making Turtle support more than single session per process....

Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by Oblivion on Fri, 28 Aug 2020 14:57:45 GMT
View Forum Message <> Reply to Message

Quote:
Well, it is still using CtrlCore and CtrlCore has a lot of global variables, so it does not really make sense to bother making Turtle support more than single session per process....

OK, thanks for the clarification.

Best regards,

Oblivion

Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by Oblivion on Mon, 31 Aug 2020 19:34:28 GMT
View Forum Message <> Reply to Message

Hello Mirek,

The Turtle package is ported to VirtualGui, and it is now in the trunk. This was the first phase. The next is to improve the server.

However, there are some small issues that might need fixing in the U++ Core, before 2020.2.

1) LaunchWebBrowser() is not working with Turtle ATM, as it uses a compile time switch (TURTLE). I could have added support for it but it had to use TURTLE switch and IMO, LaunchWebBrowser should be handled globally for VirtualGui, not just for Turtle. Otther possible/future virtualization software might want to utilize it too. (Making a callback-based version of LaunchWebBrowser for VirtualGui platform, maybe?)

2) Accordingly the TURTLE switches and Turtle specific functionality in the uppsrc seems obsolete now. VIRTUALGUI switches are sufficient. We'll already have feature-parity with the previous version of Turtle package once the LaunchWebBrowser issue is solved.

3) As it happens, I saw a Ctrl::IsEndSession() method defined in CtrlCore (always returns false). I think this method is a better way to check for session termination, at least in VirtualGui/Turtle. So I was wondering if it is possible to define it in VirtualGui/Ctrl, and let it return the status of fbEndSession flag? The reason why I ask is, it can be used to break loops that lie outside the scope of Ctrl::ProcessEvent(), in case of session termination. (i.e. Turtle's SyncClient() method is one of them...)

Best regards,
Oblivion

Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by mirek on Tue, 05 Jan 2021 10:16:06 GMT
View Forum Message <> Reply to Message

While trying to move Turtle to UppHub, I have noticed it does not work :) (I will gladly provide you commit rights there as soon as I find out your github username...)

The reason seems to be that after uploading html, we try to establish WebSocket connection over the same port. Unfortunately, at the same time browser tries to fetch favicon, which gets cought by WebSocket and connection is terminated because that is not the correct websocket request.

Have you noticed this issue?

Trying to remedy that by moving websocket communication to another port...

Mirek

Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by Oblivion on Tue, 05 Jan 2021 10:39:05 GMT
View Forum Message <> Reply to Message

Hello Mirek,

Quote:While trying to move Turtle to UppHub, I have noticed it does not work Smile (I will gladly provide you commit rights there as soon as I find out your github username...)

The reason seems to be that after uploading html, we try to establish WebSocket connection over the same port. Unfortunately, at the same time browser tries to fetch favicon, which gets cought by WebSocket and connection is terminated because that is not the correct websocket request.

Have you noticed this issue?

Trying to remedy that by moving websocket communication to another port...

Thanks for reporting this. I haven't encountered this issue. I'll look into it tonight but I am already working on a simple login mechanism, and the Turtle server side is going to change soon.
Stil, IMO it would be better to use a randomized port number for websocket in ephemeral port range or > 32K.

Speaking of login: the mainstream browsers do not allow/deprecated unsecure websocket connections unless the host is localhost or explicitly allowd. In any case using unsecure layer is bad practice. And the K.I.S.S way of implementing a login mechanism is to use the secure web socket layer (wss/TLS). (I'll propose the plan next week)

But at the moment WebSocket class seems to allow server side security via its older interface. Is there a way or plan to fix that or can I safely use the older API?

my github username: ismail-yilmaz

Best regards,
Oblivion

Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by mirek on Tue, 05 Jan 2021 15:12:53 GMT

Well, continue here please:

https://github.com/mirek-fidler/Turtle

Mirek

Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by Novo on Tue, 05 Jan 2021 15:48:14 GMT

Am I missing something?

./umk reference WebWord CLANG -bus
In file included from /home/buildbot/worker/l-upp/build/reference/WebWord/main.cpp:1:
/home/buildbot/worker/l-upp/build/reference/WebWord/WebWord.h:5:10: fatal error:
'Turtle/Turtle.h' file not found
#include <Turtle/Turtle.h>
        ^~~~~~~~~~~~~~~~~
1 error generated.

Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by mirek on Tue, 05 Jan 2021 15:52:22 GMT

Novo wrote on Tue, 05 January 2021 16:48Am I missing something?

./umk reference WebWord CLANG -bus
In file included from /home/buildbot/worker/l-upp/build/reference/WebWord/main.cpp:1:
/home/buildbot/worker/l-upp/build/reference/WebWord/WebWord.h:5:10: fatal error:
'Turtle/Turtle.h' file not found
#include <Turtle/Turtle.h>
        ^~~~~~~~~~~~~~~~~
1 error generated.

It is not there anymore... Actually, nor is WebWord.

Mirek

Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by Oblivion on Tue, 05 Jan 2021 16:56:53 GMT
View Forum Message <> Reply to Message

Hello Mirek,

There was also a rudimentary WebHelloWorld example left in the reference section. I have moved it to UppHub/Turtle/Examples directory.

Best regards,
Oblivion

Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by mirek on Tue, 05 Jan 2021 17:00:44 GMT
View Forum Message <> Reply to Message

Novo wrote on Tue, 05 January 2021 16:48Am I missing something?

./umk reference WebWord CLANG -bus
In file included from /home/buildbot/worker/l-upp/build/reference/WebWord/main.cpp:1:
/home/buildbot/worker/l-upp/build/reference/WebWord/WebWord.h:5:10: fatal error:
'Turtle/Turtle.h' file not found
#include <Turtle/Turtle.h>
       ^~~~~~~~~~~~~~~~~
1 error generated.

Speaking of which: It would be great if your automated building infrastructure started scanning UppHub. Do you think it would be possible?

Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by Novo on Tue, 05 Jan 2021 18:05:34 GMT
View Forum Message <> Reply to Message

mirek wrote on Tue, 05 January 2021 12:00
Speaking of which: It would be great if your automated building infrastructure started scanning UppHub. Do you think it would be possible?
Theoretically, yes.
What do I need to do for that?

Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by mirek on Tue, 05 Jan 2021 19:22:24 GMT
View Forum Message <> Reply to Message

Novo wrote on Tue, 05 January 2021 19:05mirek wrote on Tue, 05 January 2021 12:00
Speaking of which: It would be great if your automated building infrastructure started scanning UppHub. Do you think it would be possible?
Theoretically, yes.
What do I need to do for that?

Preliminary: Start here:

  https://github.com/ultimatepp/ultimatepp/blob/master/upphub. root

then scan through all lists, clone/pull nests, compile (or even test).

I will create some docs about the listings json format soon...

Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by Oblivion on Thu, 07 Jan 2021 09:15:55 GMT

Hello Mirek,

If you have no objections, I'd like to mark the current version of Turtle package as "technical preview" until we implement a security/login mechanism.

Best regards,
Oblivion

Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by mirek on Thu, 07 Jan 2021 12:40:42 GMT

Oblivion wrote on Thu, 07 January 2021 10:15Hello Mirek,

If you have no objections, I'd like to mark the current version of Turtle package as "technical preview" until we implement a security/login mechanism.

Best regards,
Oblivion

Sure.

Anyway, sorry for somewhat late reply, but are you sure you want to go there wrt login? I mean, originally I planned that login mechanism will be left to be implemented by client in some web langauge (or Skylark).

Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by Oblivion on Thu, 07 Jan 2021 14:16:13 GMT
View Forum Message <> Reply to Message

Hello Mirek,
Quote:are you sure you want to go there wrt login? I mean, originally I planned that login mechanism will be left to be implemented by client in some web langauge (or Skylark).

The thing is I didn't know the original design decisions and the prospect for the Turtle package, until now. :)
So I assumed that it is expected to be implemented. This is actually a good news and a reasonable approach.

I have no objections to less work on my side. :d Thus what I'd like to do now is document the simple wire protocol and add a better description to README.

As a side note: I already implemented a proof-of-concept Turtle-protocol based VirtualGui/SshGui (as ssh2 sub-system, usingo our SSH package) that effectively solves the security/login problem for U++ remote frontend.
And implemented a simple TurtleCtrl prototype that works with both html/ws and this ssh2 backends.

Best regards,
Oblivion

Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by mirek on Thu, 07 Jan 2021 14:30:01 GMT
View Forum Message <> Reply to Message

Quote:
As a side note: I already implemented a proof-of-concept Turtle-protocol based VirtualGui/SshGui (as ssh2 sub-system, usingo our SSH package) that effectively solves the security/login problem for U++ remote frontend.
And implemented a simple TurtleCtrl prototype that works with both html/ws and this ssh2 backends.

Sounds good. It is all yours now :)

Mirek

Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by Novo on Sun, 10 Jan 2021 04:28:55 GMT

mirek wrote on Tue, 05 January 2021 14:22Novo wrote on Tue, 05 January 2021 19:05mirek wrote on Tue, 05 January 2021 12:00
Speaking of which: It would be great if your automated building infrastructure started scanning UppHub. Do you think it would be possible?
Theoretically, yes.
What do I need to do for that?

Preliminary: Start here:

   https://github.com/ultimatepp/ultimatepp/blob/master/upphub. root

then scan through all lists, clone/pull nests, compile (or even test).

I will create some docs about the listings json format soon...

Initial version works.
I can read json file from internet, retrieve repo url, scan for *.upp files, and build them.
Unfortunately, the very first error I get with the package Urr is
undefined reference to `main'
I can read more settings from the json file, like to build a package or not, supported platforms and compilers.
In my own current configuration I manually support lists of packages for each platform. That includes U++.

Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by Novo on Sun, 10 Jan 2021 17:41:39 GMT

Novo wrote on Sat, 09 January 2021 23:28
I can read more settings from the json file, like to build a package or not, supported platforms and compilers.
On second thought, I won't be able to pass configuration parameters from a json file, which I read on a master node, to a worker node, which is dynamically generating build steps.

This means that I can support only two modes:
1) All projects should be buildable on all platforms with all compilers.
2) I manually support lists of build steps.

1 is much easier to support.
2 is more realistic.

Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by Novo on Tue, 12 Jan 2021 02:14:29 GMT

It would be great to add code below to packages Turtle and Skylark:
#ifdef flagMAIN
CONSOLE_APP_MAIN {
}
#endif
That would allow me to build repos Turtle and Skylark fully automatically.

TIA

Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by Oblivion on Tue, 12 Jan 2021 05:48:54 GMT

Hello Novo,
Quote:It would be great to add code below to packages Turtle and Skylark:

I have added the lines to Turtle/Init.cpp, please check.

Best regards,
Oblivion

Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by Novo on Tue, 12 Jan 2021 06:15:02 GMT

Oblivion wrote on Tue, 12 January 2021 00:48Hello Novo,
Quote:It would be great to add code below to packages Turtle and Skylark:

I have added the lines to Turtle/Init.cpp, please check.

Best regards,
Oblivion
Thanks! It would be great to do the same with Skylark.

I cannot build Turtle on Mac.
./umk .,/Users/ssg/dvlp/cpp/code/upp/git/uppsrc examples/WebHelloWorld CLANGcpp17 -bus
/Users/ssg/dvlp/cpp/code/upp/git/uppsrc/CtrlLib/CtrlUtil.h:57:7: error: redefinition of 'PrinterJob'
class PrinterJob {
      ^
/Users/ssg/dvlp/cpp/code/upp/git/uppsrc/VirtualGui/VirtualGui.h:109:7: note: previous definition is

---

here
class PrinterJob { // Dummy only...
    ^

Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by Oblivion on Tue, 12 Jan 2021 08:03:03 GMT
View Forum Message <> Reply to Message

Quote:I cannot build Turtle on Mac.
./umk .,/Users/ssg/dvlp/cpp/code/upp/git/uppsrc examples/WebHelloWorld CLANGcpp17 -bus

Is your build system reading the *upp files? Turtle examples allow GUI/TURTLEGUI switching and it seems like the TURTLEGUI flag in the package configuration file is not getting activated. This flag should bring in the CtrlLib/Turtle packages in a mutually exclusive way.)

Maybe I should remove the switch from the reference examples. (Since these switches are already available to user, with project templates.)

Edit: I have removed the switch from the examples. Please check.

Best regards,
Oblivion

Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by Novo on Tue, 12 Jan 2021 19:33:09 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Tue, 12 January 2021 03:03
Edit: I have removed the switch from the examples. Please check.

Best regards,
Oblivion
It is still broken:
/Users/ssg/dvlp/cpp/code/upp/git/uppsrc/VirtualGui/VirtualGui.h:110:22: error: expected member name or ';' after declaration specifiers; 'nullptr' is a keyword in Objective-C++
    NilDraw   nil;
    ~~~~~~~   ^
/usr/include/objc/objc.h:106:16: note: expanded from macro 'nil'
define nil nullptr
          ^

This is a MacOS problem. Theoretically, it should be detected by an Upp build. But Upp build seems to be fine.

Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by Novo on Tue, 12 Jan 2021 19:38:58 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Tue, 12 January 2021 03:03
Is your build system reading the *upp files?
Nope. I'm not parsing .upp files. I'm planning to make my own replacement of umk in the future, but this task has low priority.
Supporting of the build system is a full time job already.

Subject: Re: [Proposal] The maintainer and the future of TURTLE package
Posted by mirek on Tue, 19 Jan 2021 08:34:39 GMT
View Forum Message <> Reply to Message

Novo wrote on Tue, 12 January 2021 03:14It would be great to add code below to packages Turtle and Skylark:
#ifdef flagMAIN
CONSOLE_APP_MAIN {
}
#endif
That would allow me to build repos Turtle and Skylark fully automatically.

TIA

Is this a good idea?

I mean I understand you cannot compile without it, but in that case maybe you should just ignore non-main packages - compiling examples should do.

Or alternatively, you can detect non-main package and simply generate a project with empty CONSOLE_APP_MAIN and using the package you are testing.

Last alternative is to put something like above to the Core.

Putting it to every single package out there seems ugly.

Mirek

Subject: Re: [Proposal] The maintainer and the future of TURTLE package

Posted by Novo on Tue, 19 Jan 2021 16:09:08 GMT

View Forum Message <> Reply to Message

mirek wrote on Tue, 19 January 2021 03:34
Is this a good idea?

Definitely not a good idea, but that could've fix Skylark build quickly and easily ...
I'll make my own umk-like tool. This is a lot of pain, but I need it anyway. Can't guarantee this will
happen soon ...