Subject: While loop, sleep
Posted by sniffgriff on Thu, 26 Oct 2023 04:50:40 GMT
View Forum Message <> Reply to Message

Hello!

I am trying to run a while loop function in U++, and I want it to run indefinitely. I trigger the while loop in a button. However, when I use sleep(), the whole program waits/freezes for the duration of sleep. Is there a way to run the loop parallel? Where do I trigger the loop function? Sorry if this is obvious, but I appreciate the input/advice.

Subject: Re: While loop, sleep
Posted by koldo on Thu, 26 Oct 2023 07:03:01 GMT
View Forum Message <> Reply to Message

Hi Sniffgriff

Waiting for a button click is very simple, and you don't have to do any looping.
There is a very clear example in Examples/Button.
The program waits, but without spending CPU, and when you click on a button, a text is changed.
You can replace that action with whatever you want.

Subject: Re: While loop, sleep
Posted by Oblivion on Thu, 26 Oct 2023 09:44:19 GMT
View Forum Message <> Reply to Message

Hi,
Quote:I trigger the while loop in a button. However, when I use sleep(), the whole program waits/freezes for the duration of sleep. Is there a way to run the loop parallel?

If you really need to loop -for a long time- in any function in a U++ gui app, you need to use Ctrl::ProcessEvents(). This function processes UI input/events.

E.g.

```
while(my_condition) {
 Ctrl::ProcessEvents();
 Sleep(20)
}
```

Best regards,
Oblivion

Subject: Re: While loop, sleep
Posted by sniffgriff on Thu, 26 Oct 2023 14:14:51 GMT
View Forum Message <> Reply to Message

Hi friends, thank you for replying!

The function im trying to run is to move the mouse to a certain position on the desktop, and I want it to loop in a while:

```
 void Random_Mouse()
{

 while (1)
 {

Ctrl::ProcessEvents();

 SetCursorPos(329,654); // move mouse to this position
 Sleep(2200);
 SetCursorPos(329,784);
 Sleep(2200);

 }

}
```

I put this function in a button:

```
 MyAppWindow() {

...

 Button_Random << [=] { Random_Mouse(); };
...

}
```

Even with ProcessEvents, the program stalls for every sleep. Am I placing the function wrong?

Thank you all for your patience with me

Subject: Re: While loop, sleep
Posted by Oblivion on Thu, 26 Oct 2023 15:44:28 GMT

View Forum Message <> Reply to Message

Sleep(2200)

This is way too high for a gui to be responsive. 2200 ms -> 2.2 secs. And you have two of them in the loop. It should be at most Sleep(20).

Other than that, if you provide an example package, I can look into the problem.

Best regards,
Oblivion

Subject: Re: While loop, sleep
Posted by sniffgriff on Thu, 26 Oct 2023 17:10:54 GMT

View Forum Message <> Reply to Message

Even if I reduce the sleep to (20), the program as a whole is suspended, even though the loop is running from triggering a button. I guess I wanted the loop to be running at the same time as everything else.

I have the U++ tracking the users mouse position on the desktop, as well as the pixel color of the mouse position. I can manually move the mouse using Windows headers, basically an auto clicker Im trying to create. Move the mouse to a position, wait a moment, click, wait a moment, move to another position, etc.

Will the package include windows header files?

Subject: Re: While loop, sleep
Posted by Oblivion on Thu, 26 Oct 2023 19:03:15 GMT

View Forum Message <> Reply to Message

Ok, the following code works on win 10:

```
#include <CtrlLib/CtrlLib.h>
#include <windows.h>

using namespace Upp;

static bool sClose;

struct MyApp : TopWindow {
 Button bt;
 MyApp()
 {
  CenterScreen().SetRect(0, 0, 640, 480);
  Add(bt.SetLabel("Click me!").HCenterPos().VCenterPos());
  WhenClose << [=] { sClose = true; };
  bt << [=] {
   Point p;
   GetCursorPos(p);
   while(p.y-- > 0 && !sClose) {
    ProcessEvents();
    SetCursorPos(p.x, p.y);
                                Title("Cursor pos (y): " << AsString(p.y));
    GuiSleep(20);
   }
  };
 }
};

GUI_APP_MAIN
{
 MyApp().Run();
}
```

Edit: Example refactored.

Best regards,
Oblivion

Subject: Re: While loop, sleep
Posted by sniffgriff on Fri, 27 Oct 2023 15:34:15 GMT
View Forum Message <> Reply to Message

Hi friend thank you for that example.

I believe it kind of works? No matter how big I make the milliseconds in GuiSleep, the speed remains the same.

Subject: Re: While loop, sleep
Posted by zsolt on Fri, 27 Oct 2023 15:43:17 GMT
View Forum Message <> Reply to Message

Why don't you use Ctrl::SetTimeCallback()?

Subject: Re: While loop, sleep
Posted by Klugier on Fri, 27 Oct 2023 19:50:17 GMT

View Forum Message <> Reply to Message

Hello,

I agree with zsolt. SetTimeCallback is the way to go in your case. Currently, I am working on expanding our GUI tutorial about exemplary usage of that feature.

Here is the code:

```
#include <CtrlLib/CtrlLib.h>

using namespace Upp;

struct MyAppWindow : TopWindow {
 Rect  rect;
 Color color;

 MyAppWindow() {
  Zoomable().SetRect(0, 0, 200, 200);

  RandomizeRect();
  SetTimeCallback(-2000, [=] { OnTimer(); });
 }

 ~MyAppWindow() {
  KillTimeCallback();
 }

 void Paint(Draw& w) override {
  Size sz = GetSize();

  w.DrawRect(sz, White());
  w.DrawRect(rect, color);
 }

 void OnTimer() {
  RandomizeRect();
  Refresh();
 }

 void RandomizeRect() {
  Size sz = GetSize();

  int length = 50;
  int x = Random() % (sz.cx - length);
  int y = Random() % (sz.cy - length);
  rect = Rect(x, y, x + length, y + length);
  color = Color(Random() % 255, Random() % 255, Random() % 255);
 }
};

GUI_APP_MAIN
{
 MyAppWindow().Run();
}
```

The first parameter in this case mean interval in milliseconds. If it is less than zero it means that it will be repeated infinite number of times. If the number is positive it will be executed only once.
SetTimeCallback(-2000, [=] { OnTimer(); });

In above example, the method OnTimer() in Ctrl will be executed once per 2 seconds until cancellation. Cancellation can be done by calling KillTimeCallback().

Moreover with SetTimeCallback you shouldn't observe any UI freezes or something like that.

Regards,
Klugier

Subject: Re: While loop, sleep
Posted by sniffgriff on Sun, 29 Oct 2023 05:18:04 GMT
View Forum Message <> Reply to Message

Thank you friend, I appreciate it.

I see that the random rectangle color is run on a single frame. How would I utilize SetTimeCallback if the loop function I am running requires waiting, between steps? For example, step:1 change the rectangle color to blue, step:2 wait 2 seconds, step:3 then change the color to red, wait 4 seconds?

Subject: Re: While loop, sleep
Posted by Klugier on Sun, 29 Oct 2023 19:22:34 GMT
View Forum Message <> Reply to Message

Hello sniffgriff ,

You can have more than one TimeCallback or you can spawn one time callback from another. For example

Spawn callbacks:

```
#include <CtrlLib/CtrlLib.h>

using namespace Upp;

class MyApp : public TopWindow {
 MyApp() {
  SetTimeCallback(1000, [=] { Action1(); }); // The first parameter is positive, time callback will be executed only once
 }

 void Action1() {
  SetTimeCallback(2000, [=] { Action2(); }); // Execute Action2 after 2 seconds...
 }

 void Action2() {
  SetTimeCallback(5000, [=] { Action3(); }); // Execute Action3 after 5 seconds...
 }

 void Action3() {
  // It's over no new time callbacks will be executed
 }
};
```

Having more than one callback time callback:

```
class MyApp : public TopWindow {
 MyApp() {
  // The third parameter of SetTimeCallback is id. If id's are different there are
  // separate entities and you can schedule multiple time callbacks at once.

  SetTimeCallback(1000, [=] { Action1(); }, 0); // Execute Action1() after 1s
  SetTimeCallback(4000, [=] { Action2(); }, 1); // Execute Action2() after 4s
  SetTimeCallback(7000, [=] { Action3(); }, 2); // Execute Action3() after 7s
 }

 void Action1() {}
 void Action2() {}
 void Action3() {}
};
```

I hope it will help.

Klugier

Subject: Re: While loop, sleep
Posted by sniffgriff on Tue, 07 Nov 2023 02:54:33 GMT
View Forum Message <> Reply to Message

Thank you my friend, im going to try this and get back to you! Thank you all very much.