

Forum:[Hack Forums](#)

Thread: [Making a button clickable every 10 seconds.](#)

SIM-810273 - December 3, 2014, 6:41 am

Hey guys how are you, so I have a game, and I want a button that when you click it the button gives the character a boost (i know how to code this) but what I can't code is once they've clicked that button they cannot click the button again for a set amount of time.

tl;dr
I'm trying to create a button which once clicked cannot be clicked again for a set amount of time.

Any help would be appreciated. Thanks in advance!

pac - December 3, 2014, 6:48 am

Put a timer which enables and disables the button every 10 seconds. I believe threading would also do fine. After your button logic is done, you disable the button, add Thread.Sleep(10000) and re-enable it again.

Niggagetsbannedinunder10 - December 3, 2014, 6:53 am

Edit: im stupid and was looking in the wrong section.

Carlsson94 - December 3, 2014, 7:01 am

(Dec 3, 2014 07:48 AM)pac Wrote: [\[CITE\]](#) Put a timer which enables and disables the button every 10 seconds. I believe threading would also do fine. After your button logic is done, you disable the button, add Thread.Sleep(10000) and re-enable it again.

Except Thread.sleep would sleep the whole application just not the button.

Malware? - December 3, 2014, 7:02 am

On a new thread, also turn illegalcrossthreading off

```
while (true)
{
    Thread.sleep(10000);
    button1.enabled = !button1.enabled;
}
```

Hawk HF - December 3, 2014, 7:14 am

(Dec 3, 2014 07:48 AM)pac Wrote: [\[CITE\]](#) Put a timer which enables and disables the button every 10 seconds. I believe threading would also do fine. After your button logic is done, you disable the button, add Thread.Sleep(10000) and re-enable it again.

Using Thread.Sleep() would freeze the entire application for 10 seconds. Not just the button.

(Dec 3, 2014 08:02 AM)Venomous Net Wrote: [\[CITE\]](#) On a new thread, also turn illegalcrossthreading off

```
while (true)
{
    Thread.sleep(10000);
    button1.enabled = !button1.enabled;
```

```
}
```

There is no point in creating a new thread just for this... And your code would enable and disable the button every 10 seconds, not on the click event as OP asked for.

Just use a timer with a 10 second interval.

The Tick() event will fire after 10 seconds. Re-enable the button then.

Odus - December 3, 2014, 9:42 am

(Dec 3, 2014 08:14 AM)Hawk HF Wrote: [\[CITE\]](#) Just use a timer with a 10 second interval.

The Tick() event will fire after 10 seconds. Re-enable the button then.

Take note everyone, no need to over complicate!

Code private void button1_Click(object sender, EventArgs e)

```
{
    button1.Enabled = false;
    timer1.Start();
}
```

private void timer1_Tick(object sender, EventArgs e)

```
{
    button1.Enabled = true;
    timer1.Stop();
}
```

ShadowBoi - December 3, 2014, 10:19 am

If you guys are going to tell him to use the Thread.Sleep() function you should start by explaining how to actually make a thread then. Because else will it sleep the main thread and that will freeze the whole program.

CrypticX - December 3, 2014, 12:42 pm

Why not just

PHP Code

```
<?php
```

private void button1_Click(object sender, EventArgs e)

```
{
    //code
    button1.Enabled = false;
    Thread.Sleep(milliseconds);
    button1.Enabled = true;
}
```

Or yeah you can do it with a timer too as mentioned above.

?????? - December 3, 2014, 12:43 pm

Please dont use thread.sleep() if you really need this feature check system time. then have a while loop that says while currenttime < time + 10
buutton is unclickable

Old_Horse - December 3, 2014, 2:40 pm

The correct way to do it is:

Code var timer = new Timer { Interval = 10000 };

```
t.Tick += delegate { button1.Enabled = !button1.Enabled; };  
t.Start();
```

Agent Raigar - December 3, 2014, 2:48 pm

Quote:also turn illegalcrossthreading off
Please don't.

Also,

(Dec 3, 2014 03:40 PM)Old_Horse Wrote: [\[CITE\]](#) The correct way to do it is:

```
Code var timer = new Timer { Interval = 10000 };  
t.Tick += delegate { button1.Enabled = !button1.Enabled; };  
t.Start();
```

There isn't a correct or wrong way to do this.

Old_Horse - December 3, 2014, 2:49 pm

(Dec 3, 2014 03:48 PM)Agent Raigar Wrote: [\[CITE\]](#) Please don't.

Also,

There isn't a correct or wrong way to do this.

Okay, let me paraphrase:

The easiest and cleanest way to do it.

Corios - December 3, 2014, 3:06 pm

(Dec 3, 2014 03:48 PM)Agent Raigar Wrote: [\[CITE\]](#) Please don't.

Also,

There isn't a correct or wrong way to do this.

actually yes there is, because this will turn the button on and off every 10 seconds. Not what the OP wants. What if he doesn't click it in the 10 seconds? it'll disable it for the next 10.

Old_Horse - December 3, 2014, 3:08 pm

(Dec 3, 2014 04:06 PM)Corios Wrote: [\[CITE\]](#) actually yes there is, because this will turn the button on and off every 10 seconds. Not what the OP wants. What if he doesn't click it in the 10 seconds? it'll disable it for the next 10.

Oops, my bad. That's what you get for only reading the title. ???

Corios - December 3, 2014, 3:09 pm

(Dec 3, 2014 04:08 PM)Old_Horse Wrote: [\[CITE\]](#) Oops, my bad. That's what you get for not reading the specs ???

ya, it was written slightly weird. I mean it's not far off and it I agree with a timer in this case, but ya that would need to be fixed :P

Sicarius - December 3, 2014, 3:10 pm

Have the button set to Unclickable (not sure what the exact method is called, believe it is Enabled();) then when the timer is done have it set to be click able again.

SL x TnT - December 3, 2014, 3:14 pm

I haven't really worked with async that much, so someone can correct this if anything's wrong.

Guessing you may want to do this with more controls. This should be easier than creating a new thread for each.

PHP Code

```
<?php
private void button1_Click(object sender, EventArgs e)
{
    DisableControl(sender as Button, 5000);
}

private void button2_Click(object sender, EventArgs e)
{
    DisableControl(sender as Button, 2000);
}

private async void DisableControl(Control con, int delay)
{
    if (con != null)
    {
        con.Enabled = false;
        await Task.Delay(delay);
        con.Enabled = true;
    }
}
```