

Forum:[Hack Forums](#)

Thread: [Installed steam games.](#)

deluxe - June 4, 2019, 12:59 pm

Hey, this is my first time coding in c# i used to code in Java however it was familiar language. Anywayss you guys probably came for the source. So it should work for 32/64 bit pc.

Screenshot:

Spoiler [\(Click to View\)](#)

Dotexe - June 4, 2019, 1:02 pm

What's the point of this ?

int - June 4, 2019, 1:03 pm

Looks ok for beginner's code, not the greatest.

Some improvements:

readPath could probably be turned into a lambda (im not good at lambdas in C#)

you can replace the hardcoded path64 and path32 with:

```
path64 = Environment.GetEnvironmentVariable("ProgramFiles(x86)")
```

```
path32 = Environment.GetEnvironmentVariable("ProgramFiles")
```

deluxe - June 4, 2019, 1:05 pm

(Jun 4, 2019 01:03 PM)int Wrote: [\[CITE\]](#) Looks ok for beginner's code, not the greatest.

Some improvements:

readPath could probably be turned into a lambda (im not good at lambdas in C#)

you can replace the hardcoded path64 and path32 with:

```
path64 = Environment.GetEnvironmentVariable("ProgramFiles(x86)")
```

```
path32 = Environment.GetEnvironmentVariable("ProgramFiles")
```

Thanks, yeah i am familiar with lambda in Java no idea how its done in c#. The for loop was a too much a shock for me lol. Thanks for the tips thought.

(Jun 4, 2019 01:02 PM)Dotexe Wrote: [\[CITE\]](#) What's the point of this ?

Well, i dont know how it wasnt clear enough, but it pretty much lets you see what games has been installed in the system. You could use it in many various things. Lets say you want to know what games your friend has on his PC. And lets say that he has an game that you wanna play and you want to borrow his account ?

Dotexe - June 4, 2019, 1:09 pm

(Jun 4, 2019 01:05 PM)DELUXE Wrote: [\[CITE\]](#) *snip*

I mean yeah i know, I didn't want to sound that harsh, I know exactly what your program does, cannot consider myself as a decent developer but I've roughly 2-3 years experience around C#. Just didn't found a point in the app, but keep it up, once you get experience go for asp.net as everything nowadays is cloud based :)

deluxe - June 4, 2019, 1:13 pm

(Jun 4, 2019 01:09 PM)Dotexe Wrote: [\[CITE\]](#) I mean yeah i know, I didn't want to sound that harsh, I know exactly what your program does, cannot consider myself as a decent developer but I've roughly 2-3 years experience around C#. Just didn't found a point in the app, but keep it up, once you get experience go for asp.net as everything nowadays is cloud based :)

I am too unfamiliar with clouds. Gotta look up into them. I am still considering going full on Python and study my way into deep learning AI's. However i found more details you could use. For example i found that you can find pretty harsh details from steams folder:

Code "users"

```
{
  "11111111111111111111"
  {
    "AccountName"      "Myloginuser"
    "PersonaName"      "my steam display name"
    "RememberPassword" "1"
    "mostrecent"       "1"
    "Timestamp"        "?"
  }
}
```

You could use this in a rat or something to determine what games an install have and his username to map it out from keylogger stuff.

StarZ - June 4, 2019, 1:28 pm

Get the Installation path from the Registry:

HKEY_CURRENT_USER\Software\Valve\Steam "SteamPath"

Because, for people like myself, I have a custom installation path. This way it will grab it dynamically.

deluxe - June 4, 2019, 1:29 pm

(Jun 4, 2019 01:28 PM)StarZ Wrote: [\[CITE\]](#) Get the Installation path from the Registry:

HKEY_CURRENT_USER\Software\Valve\Steam "SteamPath"

Because, for people like myself, I have a custom installation path. This way it will grab it dynamically.

Thanks for the tip! Will update the source :)

vaiperboy - June 5, 2019, 9:29 am

you should go with reg in stuff like this
never put hardcoded stuff

Washi - June 5, 2019, 3:27 pm

(Jun 4, 2019 01:03 PM)int Wrote: [\[CITE\]](#) Looks ok for beginner's code, not the greatest.
Some improvements:

readPath could probably be turned into a lambda (im not good at lambdas in C#)

you can replace the hardcoded path64 and path32 with:

path64 = Environment.GetEnvironmentVariable("ProgramFiles(x86)")

path32 = Environment.GetEnvironmentVariable("ProgramFiles")

Not to be rude, but those are pretty bad suggestions :/

Why would turning `readPath` into a lambda function be a good idea? I think you're misunderstanding what lambdas are used for. Maybe you meant LINQ, but that's also really not applicable here either. The only thing that can be done to get rid of the loop in `readPath` would be to use [ListBox.AddRange](#) instead. `AddRange` is a lot more efficient for adding multiple items, as it allocates the space required to store the objects put into the listbox all at once, while repeatedly calling [ListBox.Add](#) will do that incrementally. (@OP: `AddRange` is similar to `addAll` in Java)

Also, why use `GetEnvironmentVariable` to get folder paths if the very same class has the [GetFolderPath](#) method and the [SpecialFolder](#) enum?

@OP:

Welcome to the world of .NET! Next to the suggestions I gave in the above, we also generally do not like to hardcode paths using the path separator `"\"` as this might differ from platform to platform. Rather, we try to use `Path.Combine` to construct a new file path (this is similar to `Paths.get("foo", "bar", "baz.txt")` in Java).

Also, try to separate UI from utilities. For instance, make `readPath` and `readFiles` completely independent from the UI, by moving them to a separate (static) class, and letting them return an `IEnumerable<string>` or one of its derivatives instead of adding the items directly. Let the methods that actually deal with the UI (such as the constructor) do the adding of the items.

Finally, `string` is a keyword in C# (whereas in Java it is not). Try to use the lowercased keyword `string` instead of directly referencing the class `String`.

You might end up with something like the following:

code ([Click to View](#))

PHP Code

```
<?php
public static class SteamGamesDetector
{
    private static string[] ReadPath(string path)
    {
        return Directory.GetDirectories(path)
            .Select(Path.GetFileName).ToArray();
    }

    public static string[] ReadFiles()
    {
        string path64 = Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.
ProgramFiles), "Steam", "steamapps", "common");
        string path32 = Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.
ProgramFilesX86), "Steam", "steamapps", "common");

        if (Environment.Is64BitOperatingSystem && Directory.Exists(path64))
        {
            return ReadPath(path64);
        }
        else if (!Environment.Is64BitOperatingSystem && Directory.Exists(path32))
        {
            return ReadPath(path32);
        }

        return Array.Empty<string>();
    }
}
```

```
}
```

```
}
```

PHP Code

```
<?php  
public partial class Form1 : Form  
{  
    public Form1()  
    {  
        InitializeComponent();  
        listBox1.Items.AddRange(SteamGamesDetector.ReadFiles());  
    }  
}
```