

As I was watching a video to help me figure out how to code movement, etc, I came across a very strange problem that I managed to figure out just fine. The problem came to me when I was trying to write a code for my player character to turn left and right depending on which direction it's moving in.

I actually somehow had trouble figuring this one out as I'm not sure whether I just have somehow typed it in the wrong place or maybe I messed with the lines in the program, but my "if" code bit was somehow not under the private void Flip() line when I ran the script to see if it was working. Which meant that not only did my character sprite not change when facing a different side, it also confused unity and visual studio and it was begging me to explain what "Flip" was, which was kind of confusing to me. However, in the end I figured out that I had just somehow misplaced the code under a different private void and after I put it in the right place it was working perfectly fine.

(I have also had to put the Flip(); method in the void update window after I have created it in the private void.)

I suppose for next time I just have to pay more attention where things are placed and be more careful with editing/adding/deleting lines after a good bulk has already been written.

The screenshot I provided is after I have fixed the issue. Sadly I didn't take a screenshot of when I was having an issue as I was a bit too interested in trying to figure out what was wrong since I was also following a video. However, I can describe it quite easily. Imagine if my code said "private void Flip()" then under it was a different chunk of code and then somewhere at the very bottom, under a different private void, my "if" statement was placed.

```
void Update()
{
    horizontal = Input.GetAxisRaw("Horizontal");
    Flip();
}

@ Unity-Nachricht | 0 Verweise
private void FixedUpdate()
{
    rb.velocity = new Vector2(horizontal * speed, rb.velocity.y);
}

1 Verweis
private void Flip()
{
    if (isFacingRight && horizontal < 0f || !isFacingRight && horizontal > 0f)
    {
        isFacingRight = !isFacingRight;
        Vector3 localScale = transform.localScale;
        localScale.x *= -1f;
        transform.localScale = localScale;
    }
}
```

Next problem I encountered was that I had trouble finding out how to create a health system even though I tried googling a lot of different ways and trying to figure out how they work, I just couldn't until I found a tutorial that finally helped.

The video I found was very useful and the person who was guiding us along was explaining things very well, however, despite that I still sometimes didn't understand everything perfectly. However, I guess that's just how it is when you're so new to programming.

I tried to figure out the last piece of code by myself but I couldn't figure it out even with watching videos and googling, so I'll just play around with the rest of the tutorial I was watching and try to make it work with the game.

As I have been following a tutorial on making Checkpoint I have come across a problem I struggle fixing. Although I have checked multiple times and followed the tutorial, I cannot get the `.GetComponent` line to work at all. I'm not sure how to fix it.

Turns out, I managed to fix it, when I was typing the code, unity on its own changed my code from, `FindGameObjectWithTag` to `FindGameObject(s)WithTag`, now that I corrected it, it seems to have fixed the issue.

```
1 using UnityEngine;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 [RequireComponent(typeof(Collider))]
6 public class CheckpointScript : MonoBehaviour
7 {
8     private Collider2D checkpointCollider;
9     private Rigidbody2D player;
10
11     void Awake()
12     {
13         checkpointCollider = GetComponent<Collider2D>();
14         player = FindObjectOfType<Player>().GetComponent<Rigidbody2D>();
15     }
16
17     // Start is called before the first frame update
18     void Start()
19     {
20     }
21
22     // Update is called once per frame
23     void Update()
24     {
25     }
26
27     void OnTriggerEnter2D(Collider2D other)
28     {
29         if (other.CompareTag("Player"))
30         {
31             player.GetComponent<Player>().CheckpointReached();
32             checkpointCollider.enabled = false;
33         }
34     }
35 }
```