

# Behaviour tutorials

```
using UnityEngine;
```

```
public class CameraFollow : MonoBehaviour {  
  
    public Transform target;  
    public float smoothSpeed = 0.125f;  
    public Vector3 offset;  
  
    void LateUpdate ()  
    {  
        Vector3 desiredPosition = target.position + offset;  
        Vector3 smoothedPosition = Vector3.Lerp (transform.position, desiredPosition, smooth  
Speed);  
        transform.position = smoothedPosition;  
  
        transform.LookAt (target);  
    }  
}
```

## Smoothed Camera

This code is used for the smoothness of the camera when it follows the characters in the game. Instead of having a fixed camera movement projected onto the character, it allows the camera to slide along with the character.

**Public transform target** – Get information about position, rotation and scale on a target

**Public float smoothSpeed** – the higher the smoothSpeed value, the faster the camera will lock onto the target, the smaller value the longer time are spent smoothing. The value is used between 0 and 1.

**Public vector3 offset** – allows the camera to be offset on all 3 axes.

**Lerp (linear Interpolation)** – process of smoothly going to one point which is point A to point B. The method that I'm using for lerp is that it takes in our current position which is point the "starting point" and then takes in our desirable position "end point" and the last which is a float called T, but it's basically any value between 0 and 1. When it's 0 it gives our first position which currently transform our position. When it's 1 it's going to give us our second position which is "desiredPosition". But if it's anything in between, it's going to give us mix of the two.

## Steps

- Select the camera and add component and create a script called "cameraFollow" in C#, once created open the script to edit it.
- Put in the following code which is "`public transform target;`" and "`public float smoothSpeed = 0.125f;`" and below it put the next following code under `void Lateupdate ()` which is `{transform.position = target.position}`
- You'll then see the two variables in the inspector, drag the "player" into the target "slot" assuming your moveable object is the player. You'll then see that the camera snaps right inside the player. Now you want to make an offset so the camera isn't inside the character.
- Apply the code `public Vector3 offset;` beneath the public float smoothspeed. So now you can change all 3 axis of the camera in the inspector. This allows you to see the whole character.
- Now place two new steps in **void LateUpdate** which is "`Vector3 desiredPosition = target.position + offset;`" and "`Vector3 smoothedPosition = Vector3.Lerp (transform.position, desiredPosition, smoothSpeed);`" replacing the previous step which was `{transform.position = target.position}`.
- Now place a new code which is `transform.position = smoothedPosition;` So what is happening now is that every frame we get the position we want to snap to, we used `Vector3.Lerp` to get a bit closer to that position and how much closer we get depends on our `smoothSpeed`.
- The final step is just making the camera to always looking at the player which is `transform.LookAt (target)`. The object we're looking at is the object.