# Unit 05d: Camera Movement



## Introduction

At the moment, when we move our player, we can move it beyond the frame and out of sight. We don't want to lose track of the player, so we need to be able to have the camera follow the player. In this unit, we'll also learn about creating easy smooth movement.

## Goal

To have the camera smoothly follow the player.

## Process

1. We need to create a new script, attached to the camera. Select the camera in the Hierarchy, and in the Inspector click the Add Component button to create a new script called `CameraFollow`.
2. Next, we'll make several class variables:

```
public class CameraFollow : MonoBehaviour
{
    public Transform player;
    public float smoothing = 0.2f;

    private Vector3 offset;
    private Vector3 velocity = Vector3.zero;
```

The `player` variable is a simple link to the player, and `smoothing` is a variable to control how the camera moves. The next two variables are used internally –– we'll see how soon.

3. From the `Start` function, we set a couple of these values:

```
    void Start()
    {
        player = GameObject.FindWithTag("Player").transform;
        offset = transform.position - player.position;
    }
```

Here, we use the `FindWithTag` method to locate the player quickly.

> If this doesn't work, make sure to check that your player has the `Player` tag assigned.

4. We now use the `LateUpdate` method to move the camera:

```
    void LateUpdate()
    {
        Vector3 targetPosition = player.position + offset;
        transform.position = Vector3.SmoothDamp(transform.position,
targetPosition, ref velocity, smoothing);
    }
```

> `LateUpdate` takes place in the same cycle as `Update`, but after all possible `Updates` have happened.

This method uses the `SmoothDamp` method, which is described as such:

```
Gradually changes a vector towards a desired goal over time.

The vector is smoothed by some spring-damper like function, which will never
overshoot. The most common use is for smoothing a follow camera.
```

This method takes the current location, and smoothly moves it to the new location. The `velocity` method gets assigned during the method, which is something we'd look into in year 2. It's *advanced*.

5. Test out the game. You may notice some jittering if the player moves quickly –– don't worry about it now, your game will still work. It's jittering because the player has engaged the physics engine, but the camera is moving during the `Update` cycle, and the two do not automatically line up. We can go through how to improve it during a later unit, if needed.

# Wrap-Up

This script does a basic job of moving the camera around to follow the player. You can use this pattern for similiar follow movement.

# Further Material