

Μετακίνηση της κάμερας

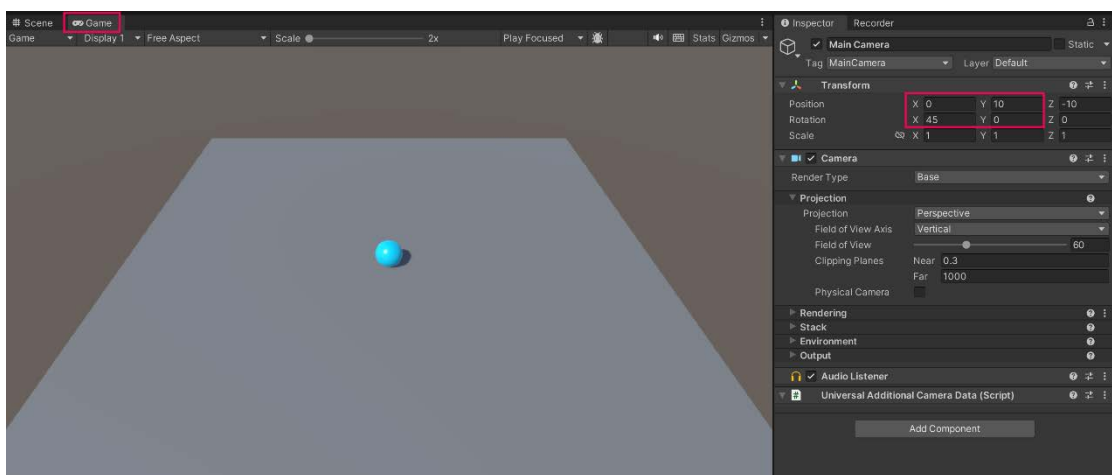
<https://learn.unity.com/tutorial/moving-the-camera?uv=6&projectId=5f158f1bedbc2a0020e51f0d#>

Σε αυτό το σεμινάριο, θα:

- Ρυθμίστε τη θέση της κάμερας
- Γράψτε ένα σενάριο για να κάνετε την Κάμερα να ακολουθεί τη σφαίρα του προγράμματος αναπαραγωγής με σταθερή μετατόπιση

2. Ρυθμίστε τη θέση της κάμερας

Ακολουθήστε τις οδηγίες βίντεο ή κειμένου παρακάτω για να σηκώσετε και να περιστρέψετε την κάμερα και, στη συνέχεια, πειραματιστείτε με το **Player** GameObject ως γονικό GameObject:



1. Σηκώστε και περιστρέψτε την κάμερα.

- Επιλέξτε την **Κύρια κάμερα** στο παράθυρο **Ιεραρχία**.
- Ορίστε την τιμή της **θέσης Y** σε περίπου **10**.
- Ορίστε την τιμή **περιστροφής X** σε περίπου **45**.
- Κάντε προεπισκόπηση της γωνίας της κάμερας στην προβολή **παιχνιδιού**.

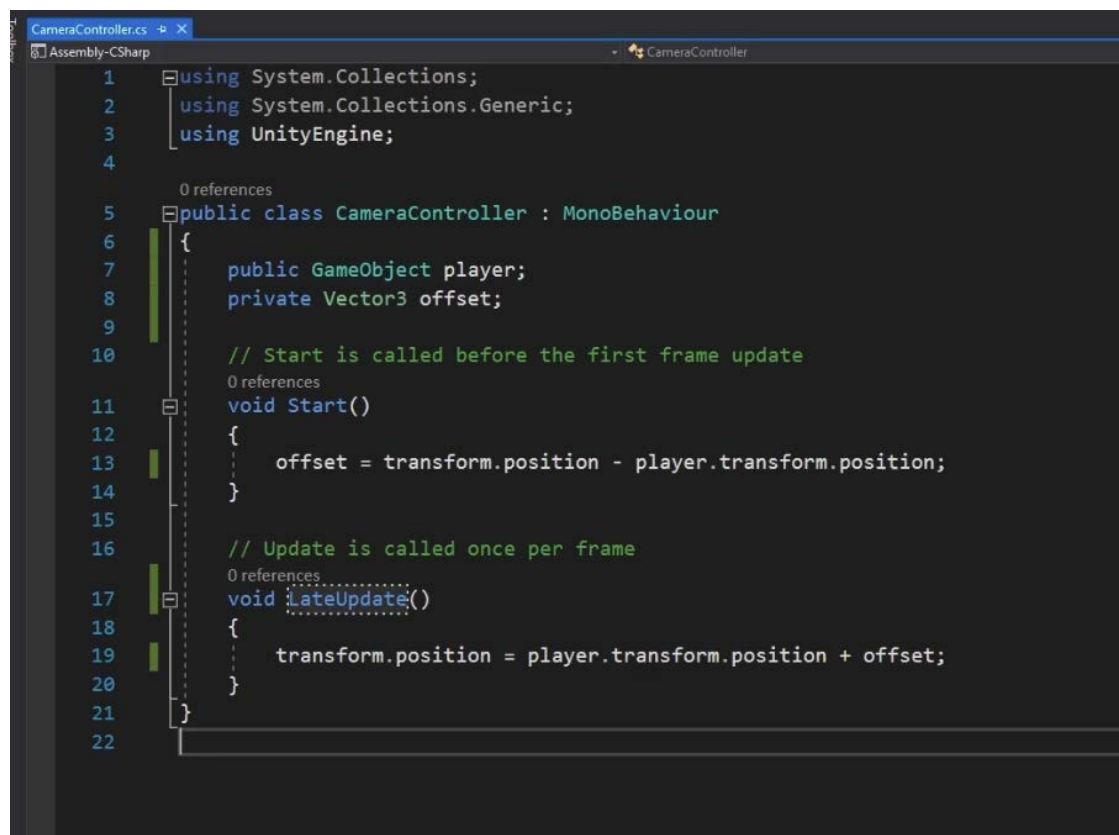
2. Πειραματιστείτε με το **Player** ως γονικό **GameObject**.

- Εάν θέλετε, σύρετε την **κύρια κάμερα** στο **Player** GameObject στο παράθυρο **Hierarchy** για να ορίσετε την κάμερα ως παιδί GameObject του **Player** GameObject.

- Δοκιμάστε το παιχνίδι και παρατηρήστε πώς η κάμερα περιστρέφεται με την κίνηση του **Player GameObject**.
- Αναιρέστε τη σχέση γονέα-παιδιού σύροντας την **Κύρια Κάμερα** πάνω από το Αντικείμενο Παιχνιδιού **Player**

3. Γράψτε ένα σενάριο CameraController

Ακολουθήστε τις παρακάτω οδηγίες βίντεο ή κειμένου για να δημιουργήσετε το σενάριο **CameraController** , να δηλώσετε τις μεταβλητές αναπαραγωγής και μετατόπισης, να ορίσετε τη μεταβλητή offset και, στη συνέχεια, να ορίσετε τη θέση της κάμερας στο **LateUpdate** :



1. Δημιουργήστε το σενάριο CameraController.

- Με επιλεγμένο το GameObject της κύριας κάμερας στο παράθυρο **Hierarchy** , επιλέξτε **Add Component > New script** στο παράθυρο **Inspector** .
- Ονομάστε το νέο σας σενάριο "CameraController".
- Στο παράθυρο **Project** , μετακινήστε το σενάριο από τον ριζικό φάκελο **Assets** στον φάκελο **Scripts** .
- Ανοίξτε το νέο σενάριο για επεξεργασία.

2. Δηλώστε μεταβλητές player και offset.

- Μέσα στο πρώτο σγουρό άγκιστρο, προσθέστε τις ακόλουθες γραμμές κώδικα για να δηλώσετε δύο νέες μεταβλητές:

```
public GameObject player;
```

```
private Vector3 offset;
```

3. Ορίστε τη μεταβλητή offset.

- Στη συνάρτηση **Start**, σε μια νέα γραμμή μετά το πρώτο σγουρό άγκιστρο, προσθέστε την ακόλουθη γραμμή κώδικα:

```
offset = transform.position - player.transform.position;
```

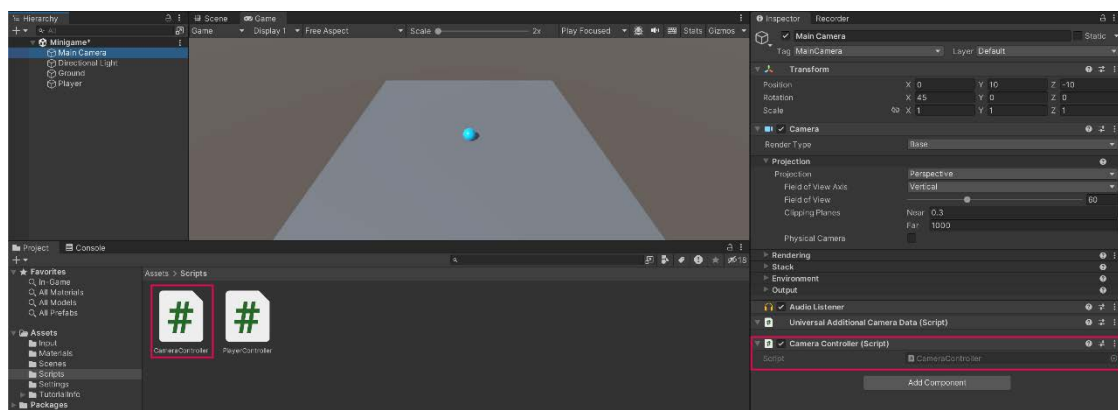
4. Ρυθμίστε τη θέση της κάμερας στο LateUpdate.

- Στη συνάρτηση Ενημέρωση, μέσα στο πρώτο σγουρό άγκιστρο, προσθέστε την ακόλουθη γραμμή κώδικα:

```
transform.position = player.transform.position + offset;
```

- Προσθέστε "Late" μπροστά από τη μέθοδο ενημέρωσης για να την αλλάξετε σε συνάρτηση LateUpdate :

```
void LateUpdate()
```

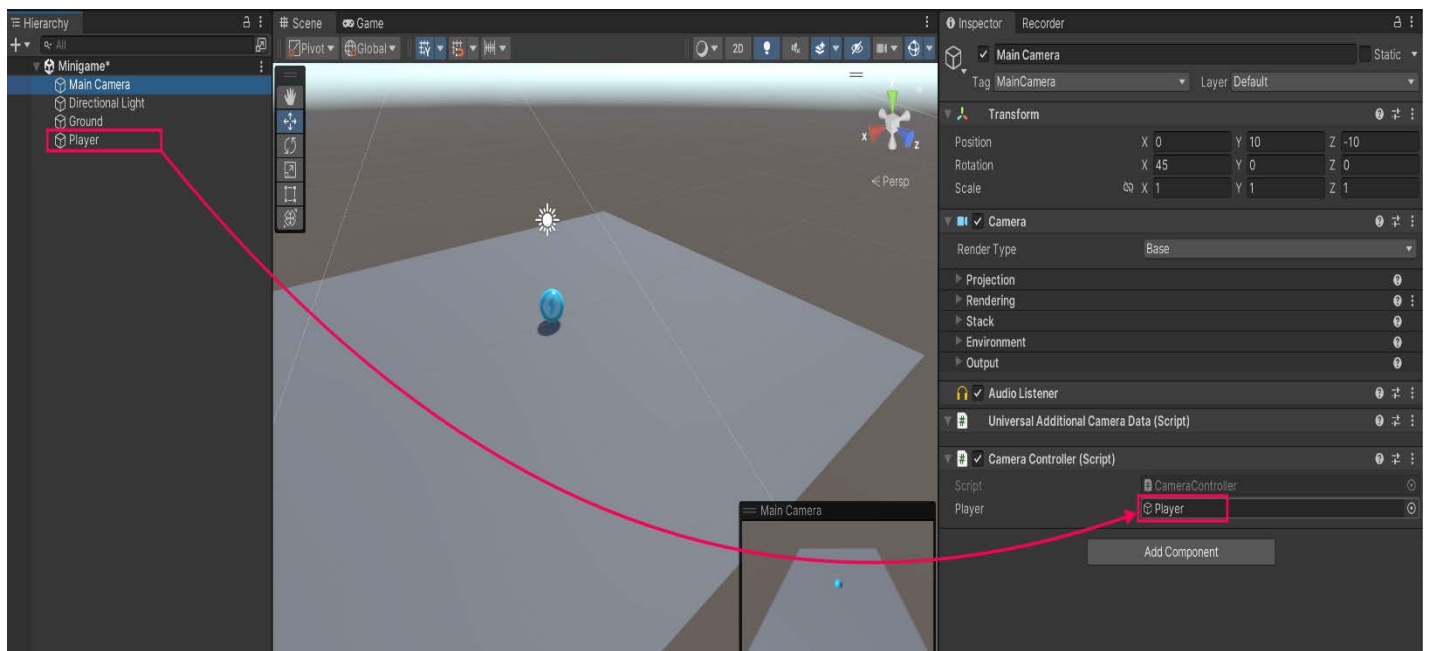


4. Αναφέρετε το Player GameObject

Ακολουθήστε τις παρακάτω οδηγίες βίντεο ή κειμένου για να αντιστοιχίσετε τη μεταβλητή **Player GameObject** στο παράθυρο του **Inspector** και, στη συνέχεια, δοκιμάστε το παιχνίδι:

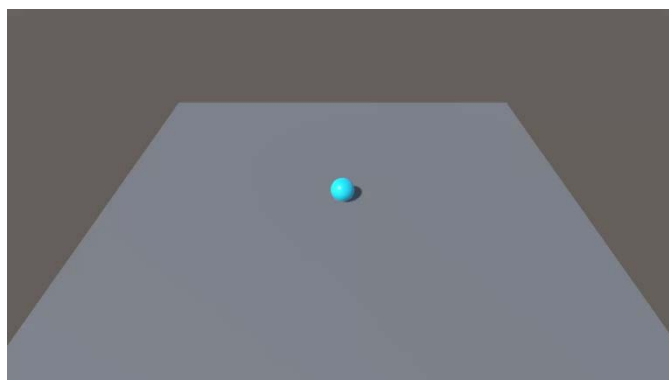
1. Αντιστοιχίστε τη μεταβλητή Player GameObject στο παράθυρο Inspector.

- Βεβαιωθείτε ότι έχετε αποθηκεύσει το σενάριο **CameraController** και, στη συνέχεια, επιστρέψτε στο Unity.
- Σύρετε το **Player GameObject** από το παράθυρο **Hierarchy** στην υποδοχή **Player** στο στοιχείο **CameraController**.



2. Δοκιμάστε το παιχνίδι.

- Η κάμερα θα πρέπει τώρα να ακολουθεί το player στην κατάλληλη μετατόπιση.



Τελικό σεναρίο

Εάν το σενάριό σας δεν λειτουργεί όπως αναμένεται, παρακάτω είναι ένα παράδειγμα του πώς θα πρέπει να μοιάζει ο κώδικάς σας σε αυτό το σημείο. Τα σχόλια έχουν προστεθεί για να γίνει ο κώδικας πιο ευανάγνωστος.

CameraController.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class CameraController : MonoBehaviour
{
    // Reference to the player GameObject.
    public GameObject player;

    // The distance between the camera and the player.
    private Vector3 offset;

    // Start is called before the first frame update.
    void Start()
    {
        // Calculate the initial offset between the camera's position
        and the player's position.
        offset = transform.position -
        player.transform.position;
    }

    // LateUpdate is called once per frame after all Update
    functions have been completed.
    void LateUpdate()
    {
        // Maintain the same offset between the camera and player
        throughout the game.
        transform.position = player.transform.position +
        offset;
    }
}
```