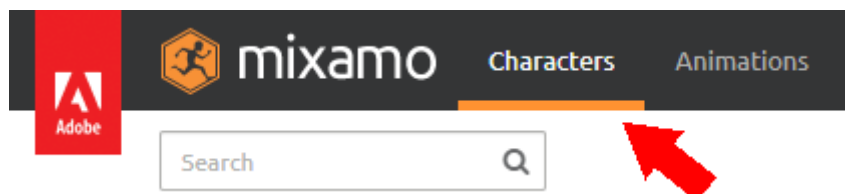


## ΜΑΘΗΜΑ 7<sup>ο</sup> – ΕΙΣΑΓΩΓΗ ΕΧΘΡΟΥ & ΔΗΜΙΟΥΡΓΙΑ ΚΙΝΗΣΕΩΝ

### Εισαγωγή εχθρού

Θέλουμε να εισάγουμε έναν εχθρό. Μεταβαίνουμε στο <https://www.mixamo.com> και από την ενότητα **Characters** διαλέγουμε έναν χαρακτήρα (π.χ. **Parasite L Starkie**).



Τον μεταφορτώνουμε στον υπολογιστή μας με τις επιλογές: **Format=Collada (.dae)** και **Pose=T-Pose**.

DOWNLOAD SETTINGS

Format	Pose
<input type="text" value="Collada(.dae)"/>	<input type="text" value="T-pose"/>

Δημιουργούμε στον υπολογιστή μας έναν φάκελο με όνομα **Zombie** και αποσυμπιέζουμε σε αυτόν τον φάκελο **textures** που υπάρχει εντός του αρχείου **Parasite L Starkie.zip**.

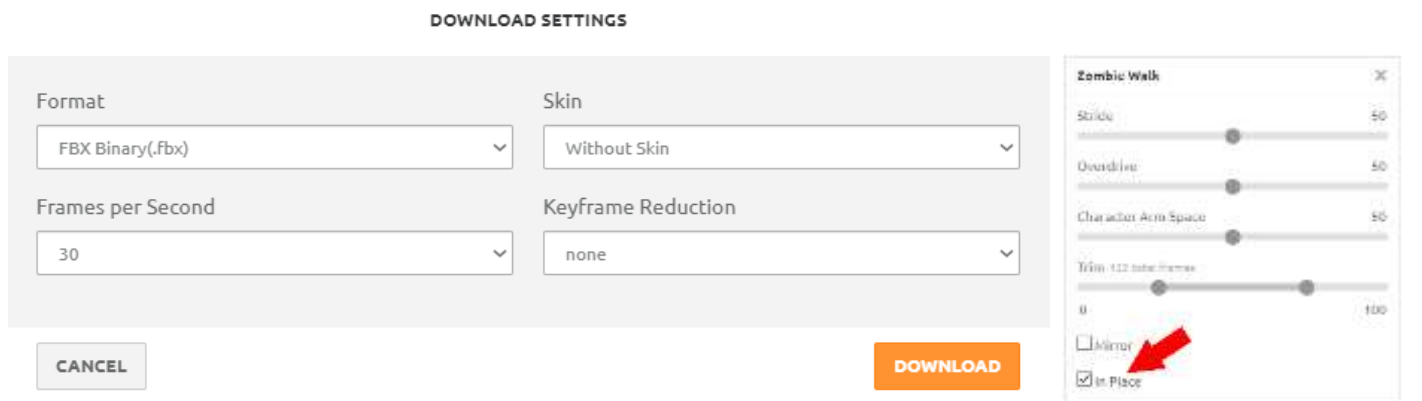
Στη συνέχεια, επιστρέφουμε στο **Mixamo** και μεταφορτώνουμε ξανά τον χαρακτήρα στον υπολογιστή μας αλλά αυτή τη φορά με **Format=FBX Binary (.fbx)** και **Pose=T-Pose**. Μετονομάζουμε το αρχείο που λάβαμε σε **Zombie.fbx** και το αντιγράφουμε στον φάκελο **Zombie**.

DOWNLOAD SETTINGS

Format	Pose
<input type="text" value="FBX Binary(.fbx)"/>	<input type="text" value="T-pose"/>

Επιστρέφουμε ξανά στο **Mixamo**, επιλέγουμε από το μενού την ενότητα **Animations**. Βρίσκουμε τέσσερις κινήσεις της αρεσκείας μας για τον χαρακτήρα μας αναζητώντας με λέξεις κλειδιά: **zombie idle**, **zombie walk**, **zombie attack**, **zombie dying**. Μεταφορτώνουμε το κάθε ένα από αυτά στον υπολογιστή μας με τις επιλογές: **Format=FBX Binary (.fbx)**,

**FPS=30, Skin=Without Skin, Keyframe=None.** Στο **Zombie Walk** ενεργοποιήστε το **In Place** στις επιλογές.



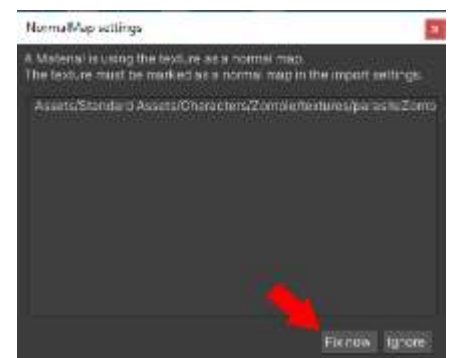
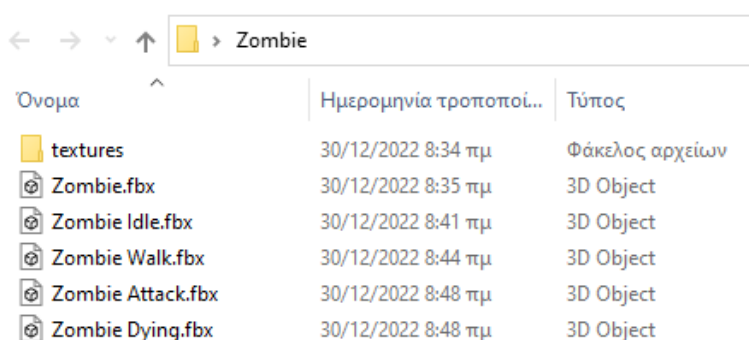
Zombie Idle.fbx

Zombie Walk.fbx

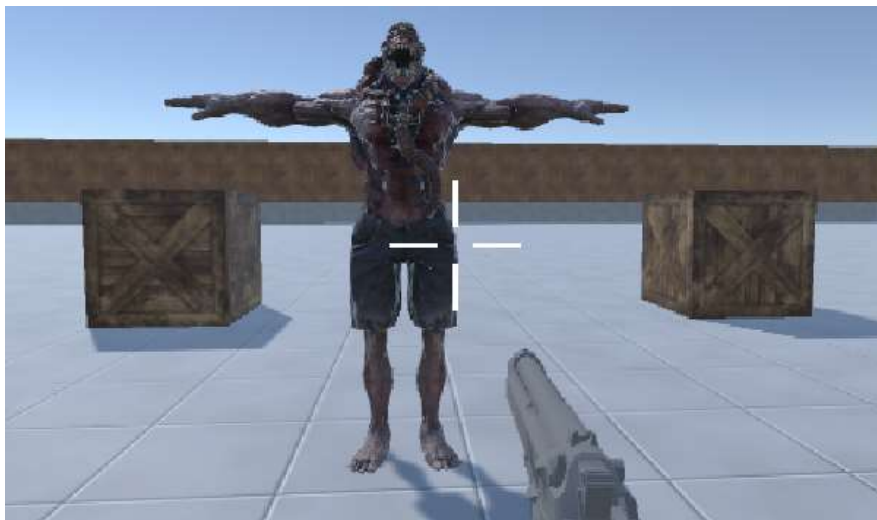
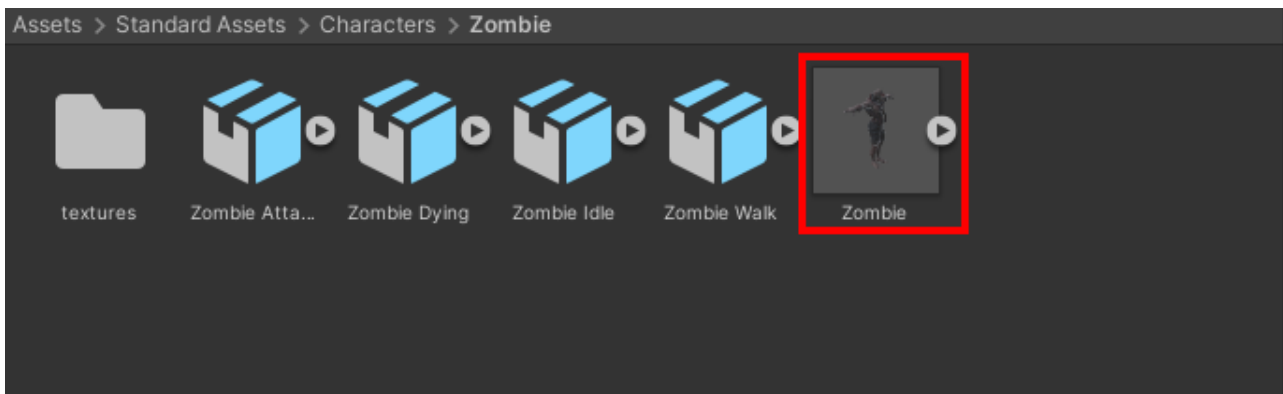
Zombie Attack.fbx

Zombie Dying.fbx

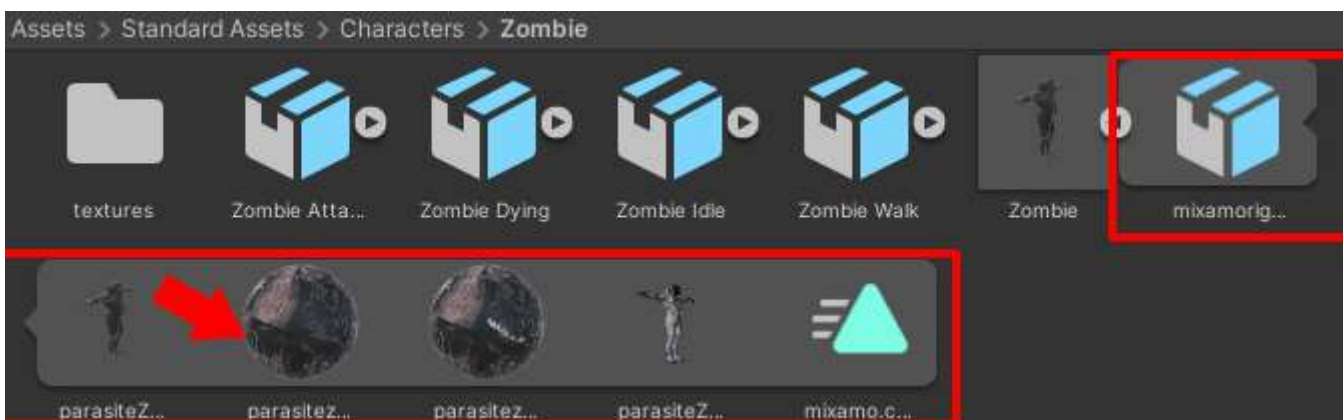
Αντιγράφουμε τα τέσσερα αρχεία **.fbx** με των κινήσεων που λάβαμε στον φάκελο **Zombie**. Κατόπιν, σύρουμε και αφήνουμε (drag-n-drop) τον φάκελο **Zombie** στον φάκελο **Characters** στο **Unity (Assets → Standard Assets → Characters)**. Πατάμε το κουμπί **Fix Now** στο μήνυμα **NormalMap settings** που θα εμφανιστεί.



Επιλέγουμε τον χαρακτήρα-εχθρό (**Zombie.fbx**) από τον φάκελο **Zombie** και τον σύρουμε στην σκηνή. Αλλάζουμε τις διαστάσεις του θέτοντας **Scale X=1.5, Y=1.5, Z=1.5**.



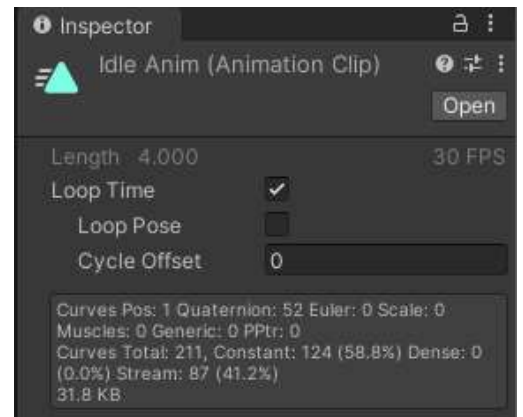
Στον φάκελο με τα αρχεία, βλέπουμε ότι δίπλα στο εικονίδιο του **Zombie.fbx** υπάρχει ένα βέλος. Αν το πατήσουμε βλέπουμε τα στοιχεία που περιέχονται εντός του αρχείου αυτού.



Κλωνοποιούμε το αρχείο με το υλικό (material) πατώντας **CTRL+D**. Στο νέο υλικό αλλάζουμε τη ρύθμιση **Rendering Mode** σε **Opaque**, το μετονομάζουμε σε **ZombieMat** και το σύρουμε επάνω στο **Zombie** εντός της σκηνής.

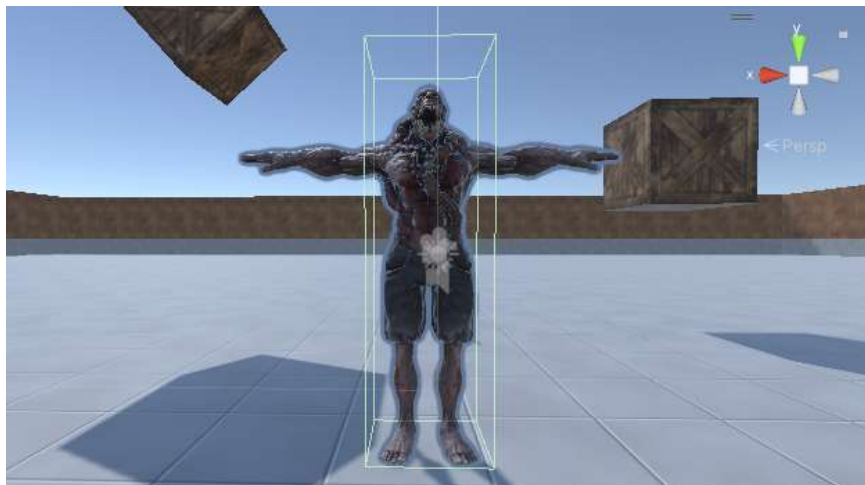
Σε κάθε ένα από τα αρχεία με τις κινήσεις (**Zombie Attack, Zombie Dying, Zombie Idle, Zombie Walk**) κάνουμε τα εξής:

- i. Πατάμε το βελάκι ( ▶ ) δίπλα στο εικονίδιο έτσι ώστε να δούμε τα περιεχόμενα.
- ii. Κλωνοποιούμε με **CTRL+D** το αρχείο με το όνομα **mixamo.com** (αρχείο κίνησης).
- iii. Μετονομάζουμε κατάλληλα το νέο αρχείο που προκύπτει (π.χ. **AttackAnim, DyingAnim, IdleAnim, WalkAnim**).
- iv. Ενεργοποιούμε το **Loop Time** σε όλες κινήσεις είναι επαναλαμβανόμενες με την πάροδο του χρόνου δηλαδή σε όλες εκτός του **DyingAnim**.



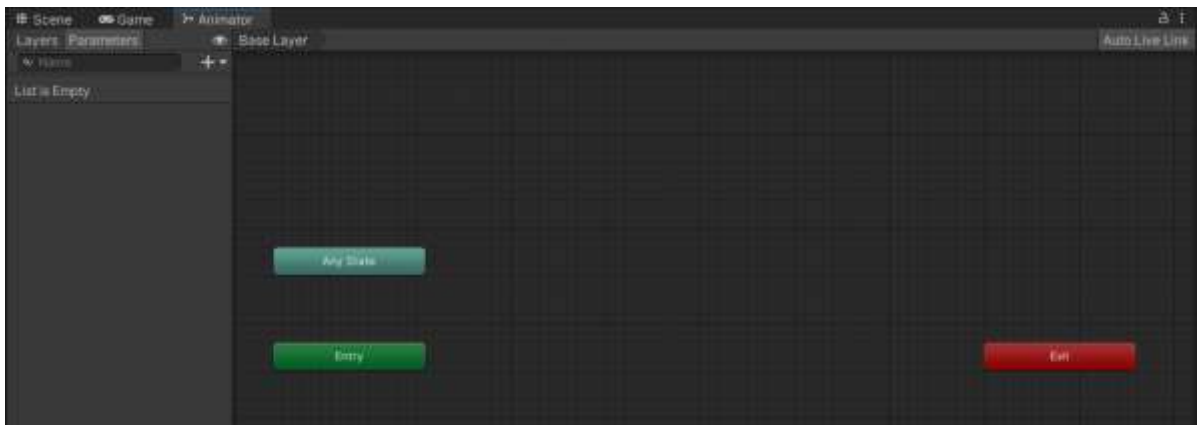
Από το παράθυρο με τις **Ιεραρχίες** επιλέγω **Zombie** και προσθέτω τα ακόλουθα εξαρτήματα:

- **Box Collider** (το διαμορφώνω για να ταιριάζει στις διαστάσεις του **Zombie** αφήνοντας τα χέρια απ'έξω).
- **Rigidbody**
- **Animator**



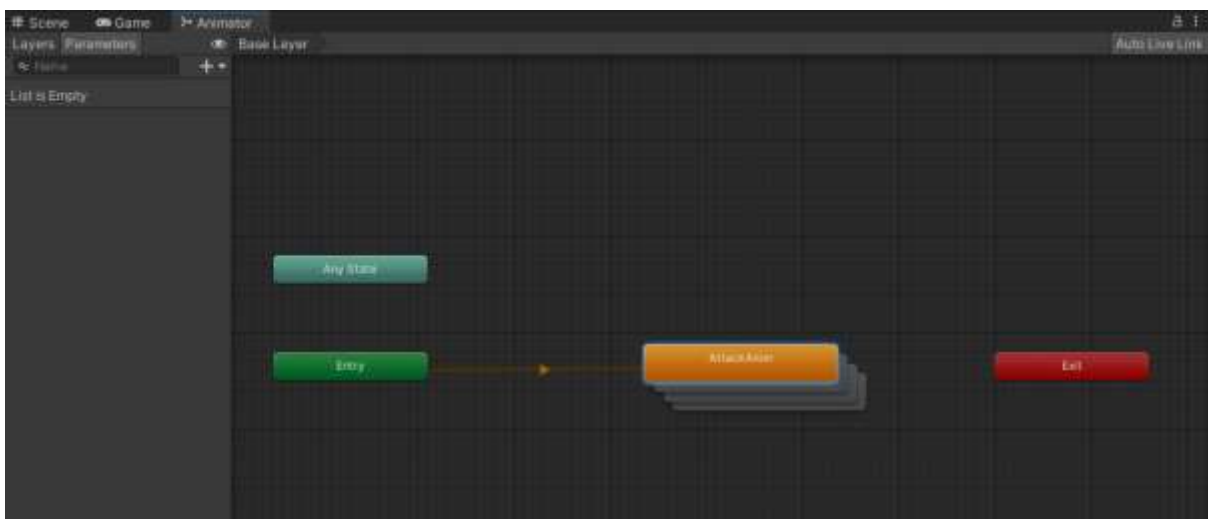
## Δημιουργία κινήσεων

Δημιουργούμε έναν ελεγκτή κινήσεων (**Animator Controller**) στον φάκελο **Animations** στα **Assets** και τον ονομάζουμε **ZombieAnimatorController**. Κάνουμε διπλό κλικ σε αυτόν και ανοίγει αυτόματα η καρτέλα **Animator**.

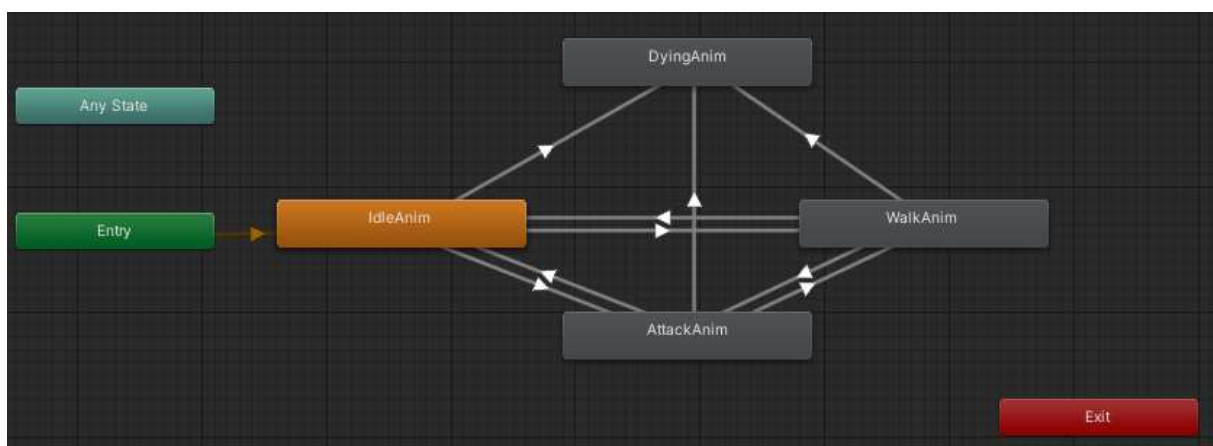


Παρατηρούμε είμαστε στη βασική στρώση κινήσεων (**Base Layer**) και εντός του πλέγματος υπάρχουν 3 σταθεροί κόμβοι: **Any State**, **Entry** και **Exit**. Θέλουμε να εισάγουμε επιπλέον τις τέσσερις κινήσεις του εχθρού (**AttackAnim**, **DyingAnim**, **IdleAnim** και **WalkAnim**) και να τους συνδέσουμε μεταξύ τους με βέλη για να δείξουμε πως μεταβαίνουμε από τη μία κίνηση στην άλλη.

Σύροντας λοιπόν και αφήνοντας στο πλέγμα τις τέσσερις κινήσεις έχουμε:



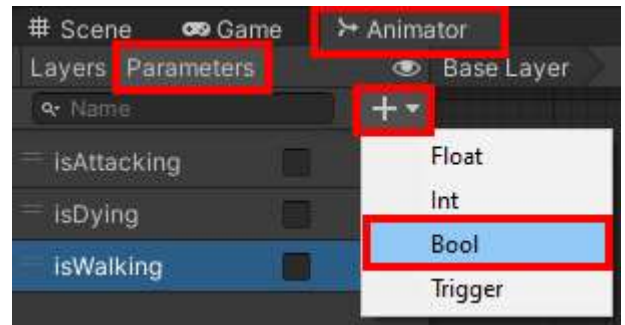
Προσπαθούμε να τακτοποιήσουμε τους νέους κόμβους σε μια λογική σειρά και να τους φέρουμε στην θέση του επόμενου σχήματος.





Συνδέουμε τους κόμβους μεταξύ τους όπως στο σχήμα χωρίς να πειράξουμε κάτι για την ώρα όσον αφορά τις επιλογές των μεταβάσεων.

Επίσης, στην καρτέλα με τις **Παραμέτρους (Parameters)** των κινήσεων δημιουργούμε τρεις λογικές μεταβλητές: **isAttacking**, **isDying**, **isWalking**. Ανάλογα με την περίπτωση, θέτοντας τις μεταβλητές αυτές είτε σε **true** είτε σε **false** θα μπορούμε να μεταβαίνουμε από την μία κίνηση στην άλλη κατά τη διάρκεια του παιχνιδιού.



Έχοντας δημιουργήσει τις τρεις λογικές μεταβλητές, επιστρέφουμε στο πλέγμα με τους κόμβους των κινήσεων. Επιλέγουμε την μετάβαση (βέλος) από το **IdleAnim** προς το **AttackAnim**.



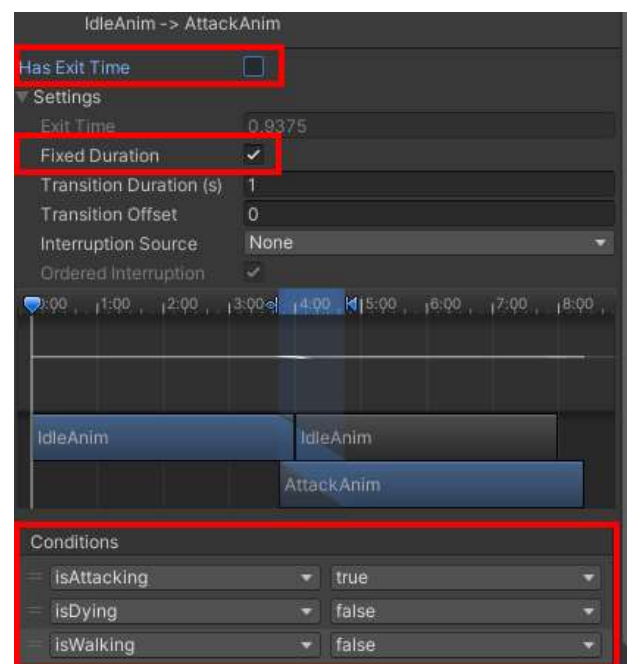
Σύρουμε το αντικείμενο **Zombie** από το παράθυρο των **Ιεραρχιών** στο παράθυρο της **Προεπισκόπησης (Preview)** που έχει εμφανιστεί κάτω δεξιά.



Κατόπιν ρυθμίζουμε για την συγκεκριμένη μετάβαση τα εξής:

- Απενεργοποιούμε το **Has Exit Time**
- **Transition Duration (s) = 1**
- Στο **Conditions** προσθέτουμε και θέτουμε:
  - isAttacking=true**
  - isDying=false**
  - isWalking=false**

Πατάμε το κουμπί **Αναπαραγωγή (Play)** στην προεπισκόπηση και ελέγχουμε την κίνηση.



Ομοίως ρυθμίζουμε και τις υπόλοιπες μεταβάσεις. Δηλαδή σε όλες:

- Απενεργοποιούμε το **Has Exit Time**
- **Transition Duration (s) = 1**

Ενώ στα **Conditions** προσθέτουμε και θέτουμε ανάλογα τις τρεις λογικές μεταβλητές. Π.χ. για την μετάβαση από **AttackAnim** σε **WalkAnim** θα έχουμε:

**isAttacking=false**

**isDying=false**

**isWalking=true**

**Σημείωση:** Αν ο κόμβος προορισμού είναι ο **IdleAnim** τότε όλες οι λογικές μεταβλητές θα έχουν την τιμή **false**.

Τέλος, από τον φάκελο **Animations** σύρουμε τον **ZombieAnimatorController** και τον αφήνουμε επάνω στο **Zombie** που βρίσκεται εντός της σκηνής.

Αν στη συνέχεια πατήσουμε **Αναπαραγωγή (Play)** και εφόσον όλα πήγαν καλά τότε θα δούμε το **Zombie** να κινείται χρησιμοποιώντας το **IdleAnim**.

## Κώδικας για την κίνηση του εχθρού

Δημιουργούμε ένα **C# Script** με όνομα **ZombieScript** στον φάκελο **Scripts** και το σύρουμε στον χαρακτήρα **Zombie** διαμέσου του παραθύρου των **Ιεραρχιών**.

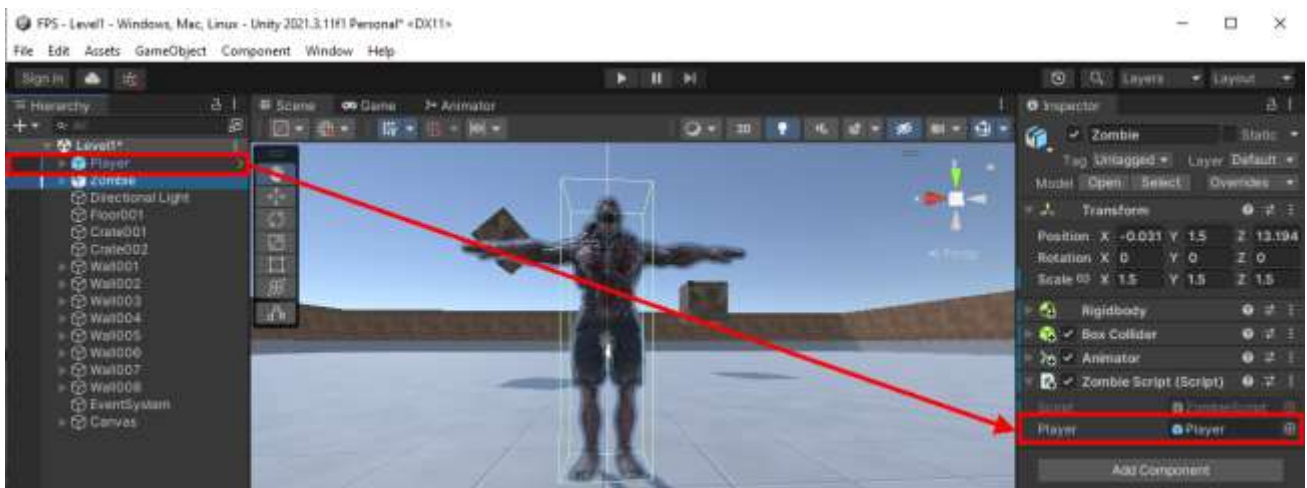
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class ZombieScript : MonoBehaviour
{
    public GameObject Player;
    private float ZombieSpeed;
    private Animator anim;
    private float DistanceFromPlayer;
    private Vector3 PlayerPosition;

    // Start is called before the first frame update
    void Start()
    {
        anim = GetComponent<Animator>();
        ZombieSpeed = 0.00f;
    }
}
```

```
// Update is called once per frame
void Update()
{
    PlayerPosition = new Vector3(Player.transform.position.x, transform.position.y, Player.transform.position.z);
    transform.LookAt(PlayerPosition);
    DistanceFromPlayer = Vector3.Distance(transform.position, Player.transform.position);
    if(DistanceFromPlayer < 2.0f)
    {
        ZombieSpeed = 0.00f;
        anim.SetBool("isWalking", false);
        anim.SetBool("isAttacking", true);
        anim.SetBool("isDying", false);
    }
    else if(DistanceFromPlayer >= 2.0f && DistanceFromPlayer <= 8.0f)
    {
        ZombieSpeed = 0.002f;
        anim.SetBool("isWalking", true);
        anim.SetBool("isAttacking", false);
        anim.SetBool("isDying", false);
        transform.position = Vector3.MoveTowards(transform.position, Player.transform.position, ZombieSpeed);
    }
    else if(DistanceFromPlayer > 8.0f)
    {
        ZombieSpeed = 0.00f;
        anim.SetBool("isWalking", false);
        anim.SetBool("isAttacking", false);
        anim.SetBool("isDying", false);
    }
}
}
```

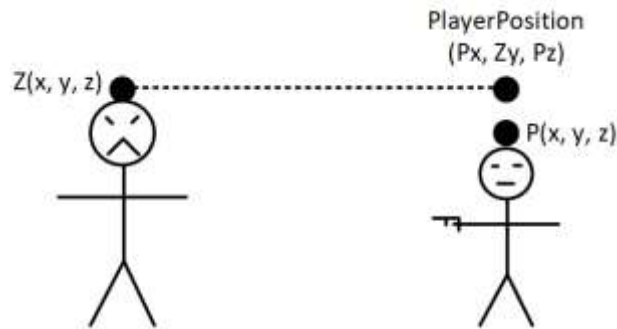
Από το παράθυρο των **Ιεραρχιών**, σύρουμε το αντικείμενο **Player** στην ιδιότητα **Player** του **ZombieScript** στο αντικείμενο **Zombie**.





Με τον παραπάνω κώδικα παρατηρούμε ότι:

1. Το Zombie κοιτάζει προς το σημείο **PlayerPosition** χρησιμοποιώντας την εντολή **transform.LookAt()**. Το σημείο αυτό χρησιμοποιεί τις συντεταγμένες X και Z του παίκτη αλλά όμως τη συντεταγμένη Y του Zombie. Αυτό συμβαίνει για να μην «χαμηλώνει» το βλέμμα του προς τον παίκτη και δημιουργεί δυσκολίες στην κίνησή του αργότερα.



2. Η απόσταση δύο σημείων υπολογίζεται εύκολα με την εντολή **Vector3.Distance()**. Αν η απόσταση του εχθρού από τον παίκτη είναι **λιγότερο από 2 μονάδες** τότε ενεργοποιείται η κίνηση της επίθεσης. Αν η απόσταση είναι **από 2 έως και 8 μονάδες** τότε ενεργοποιείται η κίνηση του βαδίσματος αλλά ταυτόχρονα ο εχθρός μετακινείται και προς τον παίκτη με την εντολή **Vector3.MoveTowards()**. Τέλος, αν η απόσταση είναι μεγαλύτερη από **8 μονάδες** τότε ο εχθρός σταματάει και βρίσκεται σε αδράνεια. Ο έλεγχος των κινήσεων γίνεται κάθε φορά θέτοντας κατάλληλα τις λογικές μεταβλητές **isAttacking**, **isDying** και **isWalking** του **ZombieAnimationController**.

Η σελίδα αυτή έμεινε σκόπιμα κενή