

Δημιουργία 2D Platform

ΑΛΛΑΓΕΣ & ΔΙΟΡΘΩΣΕΙΣ ΣΤΗΝ ΚΙΝΗΣΗ ΤΟΥ ΠΑΙΚΤΗ – ΝΕΑ ANIMATION

ΣΤΕΛΛΑ ΚΑΙΝΟΥΡΓΙΑΚΗ
2022-2023

Προσθήκη ταχύτητας στην κίνηση του παίκτη

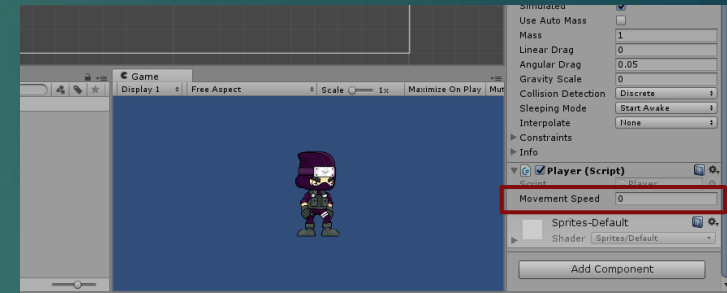
- ▶ Δημιουργούμε στον κώδικα μια μεταβλητή και για να έχουμε πρόσβαση μέσα από τον Inspector συμπληρώνουμε το πεδίο SerializeField.

[SerializeField]

```
private float movementSpeed;
```

- ▶ Μπορούμε να αλλάξουμε από εδώ την ταχύτητα αλλά για να αλλάξει κάτι και στον Player θα χρειαστεί να πολλαπλασιάσουμε την κίνηση του με την ταχύτητα.

```
myRigidbody.velocity = new Vector2(horizontal * movementSpeed,  
myRigidbody.velocity.y);
```



Προσθήκη ταχύτητας στην κίνηση του παίκτη

- ▶ Τώρα μπορούμε να δοκιμάσουμε το Play Mode και να αλλάξουμε το νούμερο στο Movement Speed για να δούμε πως αλλάζει η ταχύτητα με την οποία κινείται ο παίκτης.
- ▶ Σε αυτό το σημείο θα πρέπει να διορθώσουμε και κάτι ακόμα. Ένας υπολογιστής μπορεί να τρέχει 60fps ενώ ένας άλλος 100fps. Στον δεύτερο υπολογιστή ο παίκτης θα κινείται πιο γρήγορα, οπότε θα πρέπει κάπως να σταθεροποιήσω την Update γράφοντας `void FixedUpdate()`

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

public class Player : MonoBehaviour {

    private Rigidbody2D myRigidbody;
    [SerializeField]
    private float movementSpeed;

    // Use this for initialization
    void Start ()
    {
        myRigidbody = GetComponent<Rigidbody2D>();
    }

    // Update is called once per frame
    void FixedUpdate ()
    {
        float horizontal = Input.GetAxis("Horizontal");
        HandleMovement(horizontal);
    }
    private void HandleMovement(float horizontal)
    {
        myRigidbody.velocity = new Vector2(horizontal*movementSpeed,
myRigidbody.velocity.y);
    }
    myRigidbody.velocity = new Vector2(horizontal*movementSpeed,
myRigidbody.velocity.y);
    }
}
```

Δοκιμάζουμε το
Play Mode –
ολοκληρωμένος
κώδικας

Περιστροφή του Player ανάλογα με τη φορά της κίνησης

- ▶ Θα πρέπει να προσθέσουμε στο script μια μεταβλητή Boolean που θα έχει να κάνει με το αν ο παίκτης κοιτάζει προς τα δεξιά (όταν κοιτάζει δεξιά θα είναι true)

```
private bool facingRight;
```

- ▶ Θα δημιουργήσουμε μια συνάρτηση η οποία βασίζεται στο ότι αν αλλάξουμε το X στο scale του Player από 1 σε -1, ο Player αλλάζει φορά
- ▶ ορίζουμε εξ αρχής το facingRight σε true (μέσα στη start)

Περιστροφή του Player ανάλογα με τη φορά της κίνησης – ορισμός συνάρτησης

```
private void Flip(float horizontal)
if (horizontal>0 && !facingRight || horizontal<0 && facingRight)
{
facingRight = !facingRight; Vector3 theScale=transform.localScale;
theScale.x*=-1; // πολλαπλασιάζουμε με -1
transform.localScale=theScale;
}
```

▶ Το σκεπτικό είναι ότι δημιουργούμε μια if η οποία ενεργοποιείται όταν ο Player κινείται προς τα δεξιά και δεν κοιτάζει δεξιά ή το αντίστροφο.

-- ανακτούμε το scale του Player → Vector3
theScale=transform.localScale;

-- πολλαπλασιάζουμε με -1 → theScale.x*=-1;

-- το τοποθετούμε στο scale του Player →
transform.localScale=theScale;

▶ Τέλος προσθέτουμε στη Update

Flip(horizontal);

```

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

public class Player : MonoBehaviour {

    private Rigidbody2D myRigidbody;
    [SerializeField]
    private float movementSpeed;
    private bool facingRight;

    // Use this for initialization
    void Start ()
    {
        facingRight = true;
        myRigidbody = GetComponent<Rigidbody2D>();
    }

    // Update is called once per frame
    void FixedUpdate ()
    {
        float horizontal = Input.GetAxis("Horizontal");
        HandleMovement(horizontal);
        Flip(horizontal);
    }
    private void HandleMovement(float horizontal)
    {
        myRigidbody.velocity = new Vector2(horizontal*movementSpeed,
myRigidbody.velocity.y);
    }
    private void Flip(float horizontal)
    {
        if (horizontal > 0 && !facingRight || horizontal < 0 && facingRight)
        {
            facingRight = !facingRight;
            Vector3 theScale = transform.localScale;
            theScale.x *= -1;
            transform.localScale = theScale;
        }
    }
}

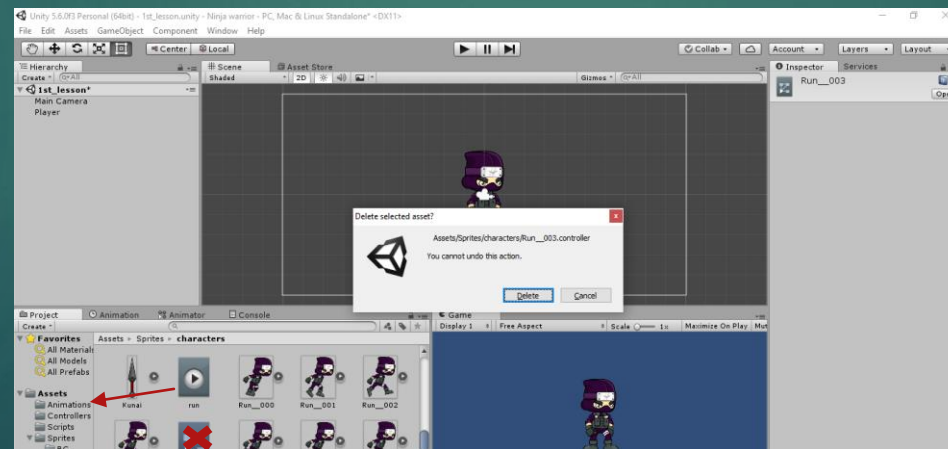
```

ΔΟΚΙΜΑΖΟΥΜΕ
ΤΟ PLAY MODE

ολοκληρωμένος
κώδικας

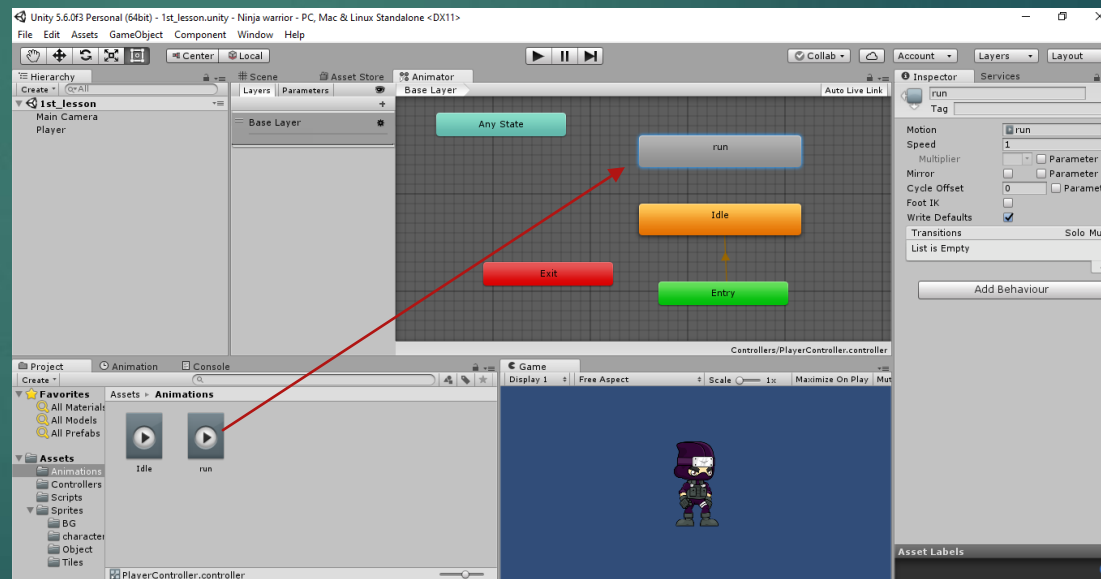
Προσθήκη νέου animation - RUN

- ▶ Θα δημιουργήσουμε ένα animation για το run με τον ίδιο τρόπο που δημιουργήσαμε για το idle. Επιλέγουμε όλες τις εικόνες runxx και τις σύρουμε στη ιεραρχία. Αυτή τη φορά όμως θα διαγράψουμε το νέο αντικείμενο που δημιουργείται στην ιεραρχία όπως και τον controller που δημιουργείται (έχουμε ήδη Player & Controller). Ονομάζουμε το νέο animation run και μετακινούμε το animation στον αντίστοιχο φάκελο.



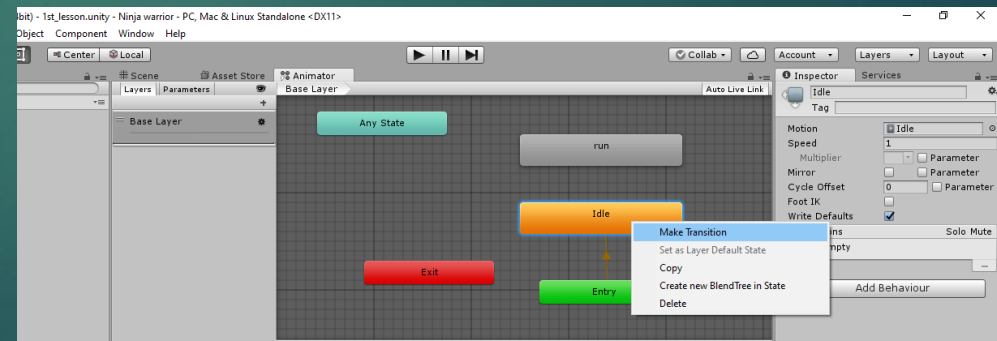
Προσθήκη νέου animation - RUN

- ▶ Θα πρέπει να προσθέσουμε το νέο animation στον Animator. Ανοίγουμε τον Animator και θα σύρουμε το νέο animation



Προσθήκη νέου animation – RUN – Δημιουργία transition

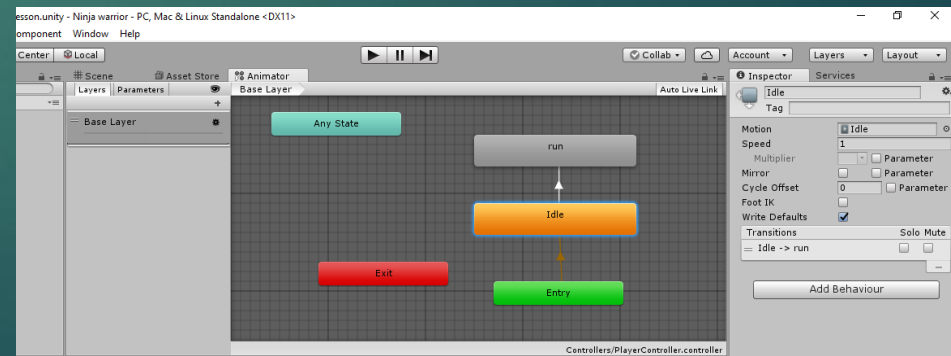
Ο Player δε γίνεται να βρίσκεται συνέχεια στην κατάσταση idle. Θα πρέπει να δημιουργήσουμε μεταβάσεις (transition) από τη μία κατάσταση στην άλλη έτσι ώστε ο παίκτης να μπορεί να μεταβεί από την κατάσταση idle στην κατάσταση run και το αντίστροφο. Για να δημιουργήσουμε ένα transition κάνουμε δ.κλικ στο αρχικό animation και επιλέγουμε Make Transition. Παρακάτω δημιουργούμε ένα transition από την κατάσταση idle στην κατάσταση run.



Προσθήκη νέου animation – RUN – Δημιουργία transition

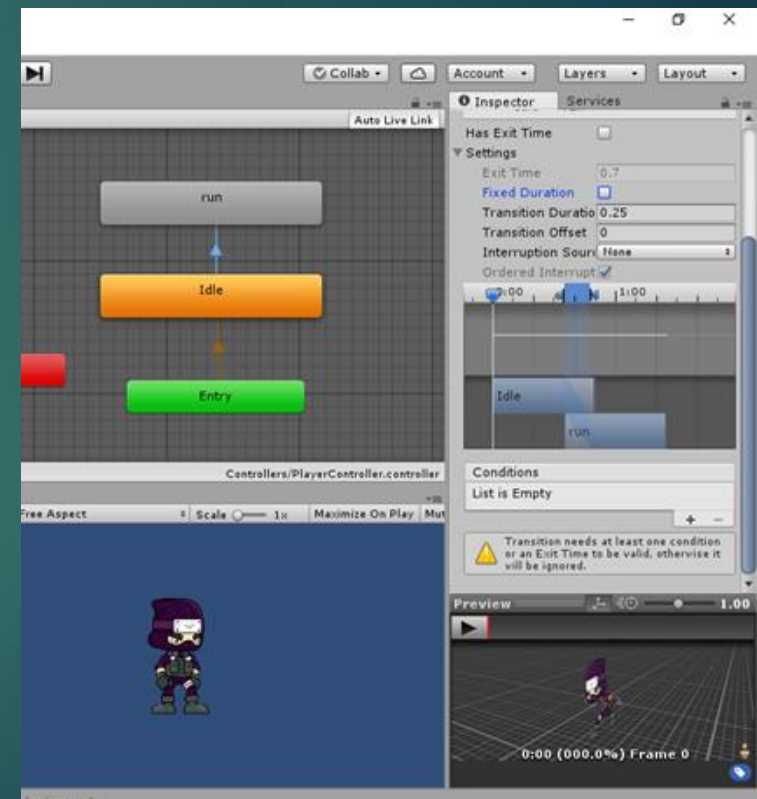
Μόλις πατήσουμε το Make Transition πατάμε πάνω στο animation στο οποίο θα καταλήγει το transition (στην περίπτωση μας στη run). Το transition φαίνεται παρακάτω (είναι με άσπρο βελάκι).

Αν τώρα πατήσουμε το Play Mode δε θα γίνει κάτι. Για να μπορέσει ένα transition να λειτουργεί, θα πρέπει να θέσουμε κάποιους όρους και να ρυθμίσουμε κάποιες παραμέτρους.



Προσθήκη νέου animation – RUN – Δημιουργία transition

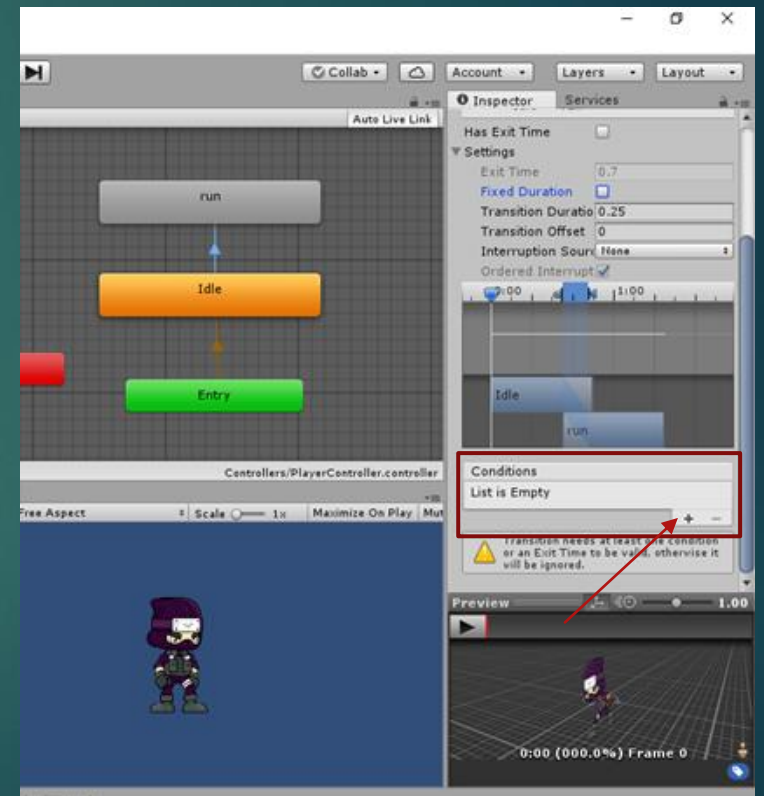
Αν επιλέξουμε το άσπρο βελάκι (του transition) θα δούμε στον inspector διάφορες παραμέτρους. Από-επιλέγουμε το has exit time και το fixed duration. Κάνουμε επίσης 0 το transition duration. Με αυτές τις ρυθμίσεις δε θα υπάρχει μετάβαση από το idle στο run animation από μόνη της, ενώ δε θα έχουμε καθόλου χρόνο αναμονής για τη μετάβαση από τη μία κατάσταση στην άλλη (όταν αυτή θα γίνει).



Προσθήκη νέου animation – RUN – Parameters & Conditions

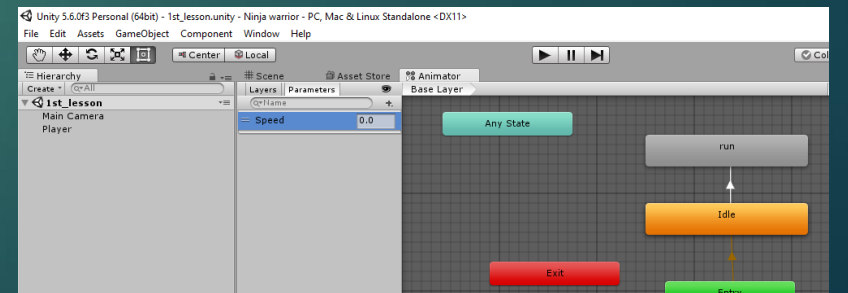
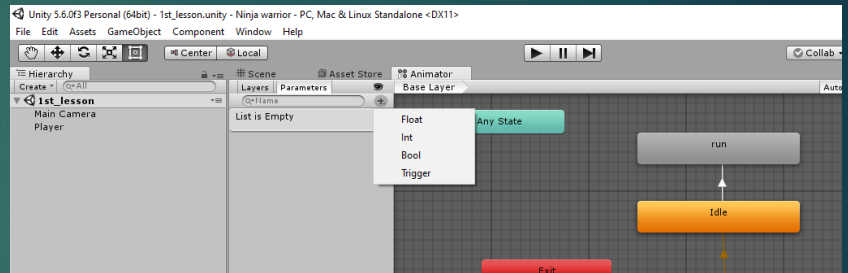
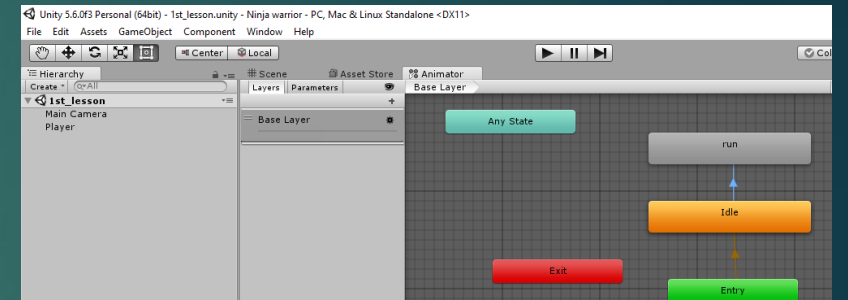
Τώρα θα πρέπει να προσθέσουμε τη λειτουργικότητα και τους όρους με τους οποίους θέλουμε να πραγματοποιείται το transition(πότε δηλαδή θα περνάμε από την κατάσταση idle στη run). Πατάμε πάνω στο βελάκι και στον inspector βρίσκουμε την ενότητα conditions.

Πατώντας το + μπορούμε να ορίσουμε όρους αλλά για να το κάνουμε αυτό θα χρειαστεί πρώτα να ορίσουμε κάποιες παραμέτρους.



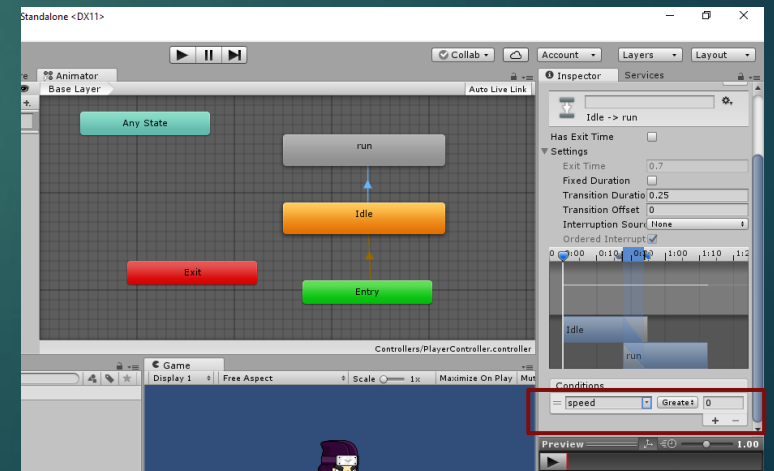
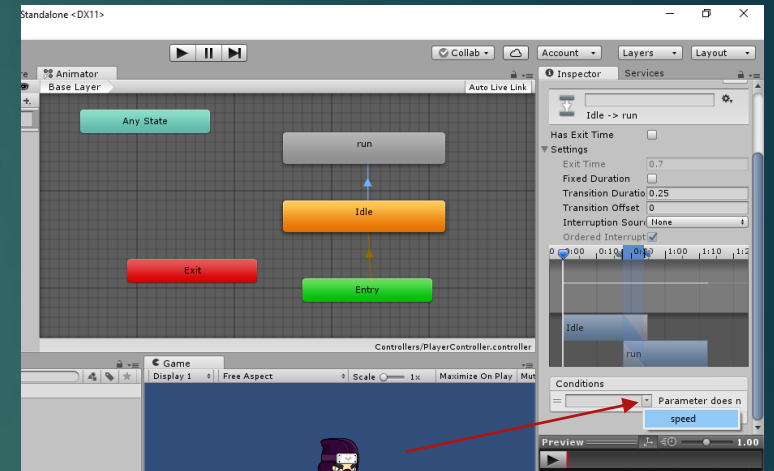
Προσθήκη νέου animation – RUN – Parameters & Conditions

- ▶ Οι παράμετροι ορίζονται στον animator στην καρτέλα Parameters
- ▶ Για να ορίσουμε μια νέα παράμετρο πατάμε το πλήκτρο + και μπορούμε να προσθέσουμε παραμέτρους διαφόρων τύπων (float, int, bool, trigger). Για την ώρα επιλέγουμε float γιατί η ταχύτητα στο πρόγραμμα μας είναι τύπου float. Την ονομάζουμε speed.



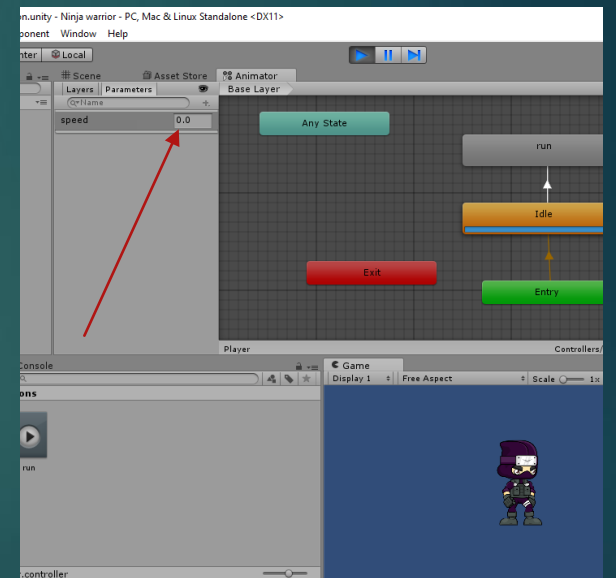
Προσθήκη νέου animation – RUN – Parameters & Conditions

- ▶ ΠΡΟΣΟΧΗ! Ότι ονομασία δώσετε στην παράμετρο, θα πρέπει να τη θυμάστε γιατί είναι case sensitive και θα χρησιμοποιηθεί στον κώδικα αργότερα!!
- ▶ Τώρα που έχουμε δημιουργήσει μια παράμετρο θα μπορούμε να την επιλέξουμε από την ενότητα Conditions. Με επιλεγμένο το βελάκι της μετάβασης πηγαίνω στην ενότητα conditions και επιλέγω τη speed.



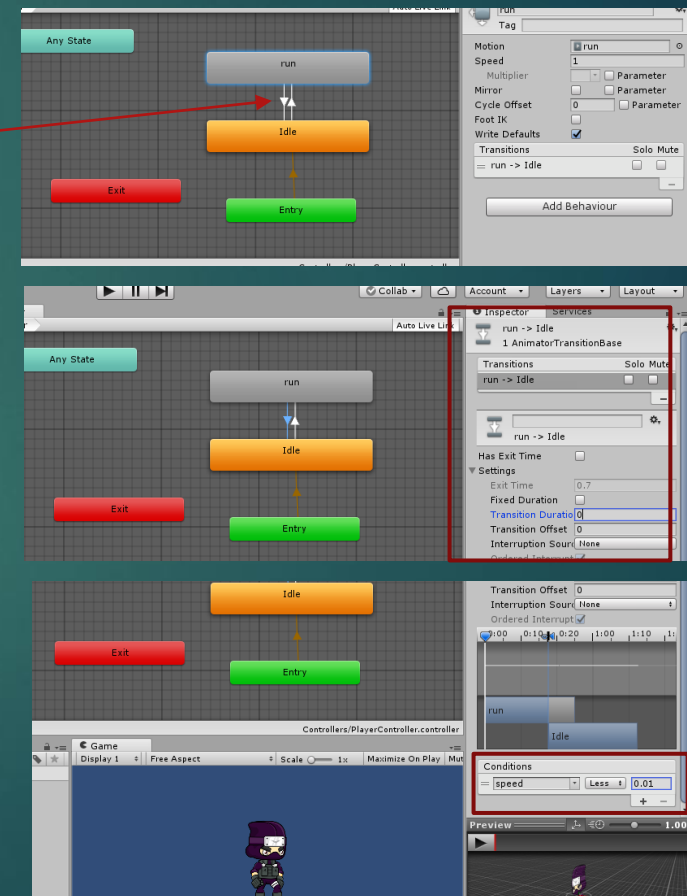
Προσθήκη νέου animation – RUN – Parameters & Conditions

- ▶ Τώρα θα πρέπει να δώσουμε τον όρο / προϋπόθεση με βάση την οποία θα γίνεται αυτή η μετάβαση. Να πούμε για παράδειγμα, πάνω από ποια ταχύτητα κίνησης ο παίκτης θα πρέπει να φαίνεται ότι τρέχει (να παίζει το run animation). Στην περίπτωση μας ορίζουμε ότι ο παίκτης θα φαίνεται ότι τρέχει αν η ταχύτητα είναι μεγαλύτερη από 0.01.
- ▶ Δοκιμάστε όσο είστε σε Play mode να αλλάξετε το speed από τις παραμέτρους από 0 σε 1 και θα δείτε ότι ο παίκτης θα μεταβεί στο animation run.



Προσθήκη νέου animation – RUN – Δημιουργία αντίστροφου transition

- ▶ Τώρα θα πρέπει να δημιουργήσουμε και το αντίστροφο transition από το run στο idle.
- ▶ Και εδώ από-επιλέγουμε το has exit time, το fixed duration και θέτουμε το transition duration σε 0.
- ▶ Το condition που θα θέσουμε για αυτό το transition θα είναι το αντίστροφο (το speed να είναι μικρότερο από 0.01)



Προσθήκη νέου animation – RUN – script

- ▶ Τώρα θα πρέπει να αλλάξουμε τον κώδικά μας, έτσι ώστε όταν ο παίκτης κινείται να ενεργοποιείται το animation run.
- ▶ Θα πρέπει να δημιουργήσουμε μια αναφορά στο component Animator.

`private Animator myAnimator;` → εδώ δημιουργούμε τη μεταβλητή
`myAnimator=GetComponent<Animator>();` → εδώ δημιουργούμε μια αναφορά στο component Animator

- ▶ μέσα στη `HandleMovement` προσθέτουμε
`myAnimator.SetFloat("speed", Mathf.Abs(horizontal));`

ΔΟΚΙΜΑΣΤΕ ΜΕ ΤΟ PLAY MODE