



Δημιουργία 2D Platform

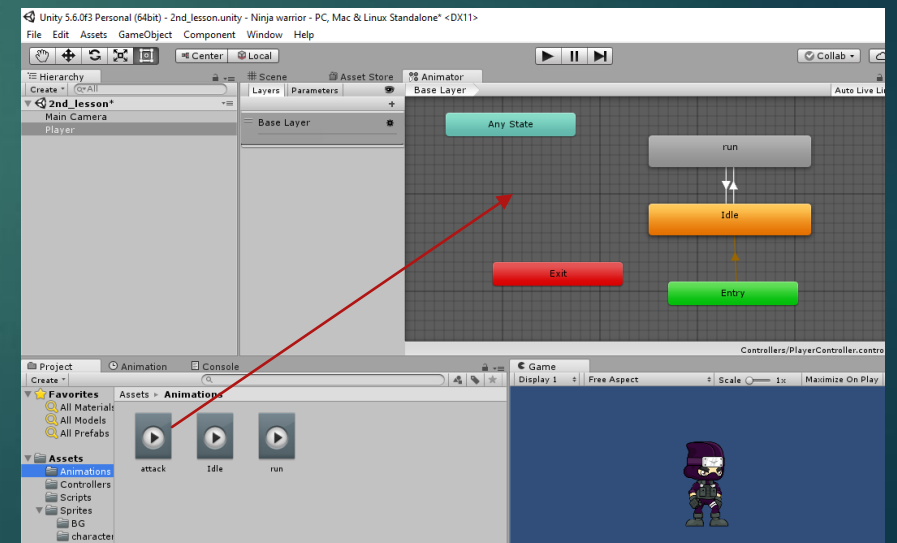
ΑΛΛΑΓΕΣ & ΔΙΟΡΘΩΣΕΙΣ ΣΤΗΝ ΚΙΝΗΣΗ ΤΟΥ ΠΑΙΚΤΗ – ΝΕΑ ANIMATION

ΣΤΕΛΛΑ ΚΑΙΝΟΥΡΓΙΑΚΗ
2022-2023

Προσθήκη animation επίθεσης - Attack

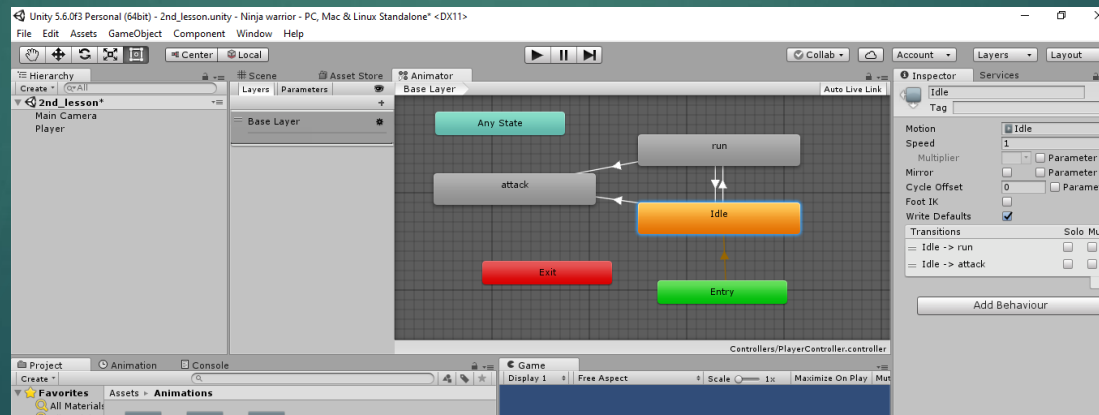
- ▶ Με τον ίδιο τρόπο που έχουμε κάνει όλα τα προηγούμενα animation θα δημιουργήσουμε ένα animation επίθεσης. Τα αντίστοιχα sprite είναι οι εικόνες attack και θα ονομάσουμε το animation attack. Διαγράφουμε τον controller και το στοιχείο που εμφανίζεται στην ιεραρχία και μεταφέρουμε το animation Attack στο φάκελο με τα animation.

Τώρα θα μεταφέρουμε το νέο animation στον animator και θα κάνουμε τις αντίστοιχες μεταβάσεις. Άρα θα πρέπει να σύρουμε το νέο animation στον Animator (όλα αυτά με επιλεγμένο τον Player)



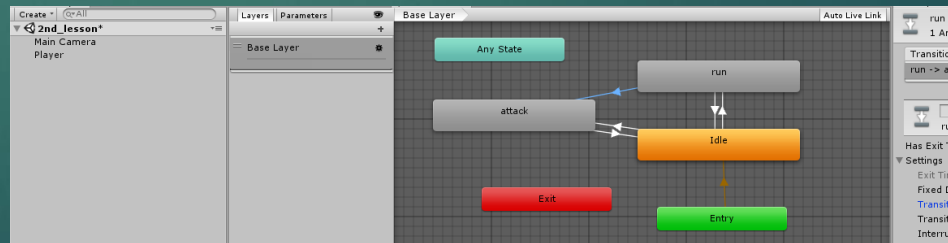
Ρυθμίσεις & transition του animation Attack

- ▶ Θα πρέπει να είμαστε σε θέση να μεταφερθούμε από το run στο attack έτσι ώστε να μπορεί να μεταβεί στην κίνηση της επίθεσης από την κίνηση run με το πάτημα του πλήκτρου που θα υποδηλώνει την επίθεση. Θα πρέπει επίσης να μπορεί να μεταβεί από την κατάσταση idle στην attack. Άρα δημιουργούμε 2 transition όπως φαίνονται παρακάτω



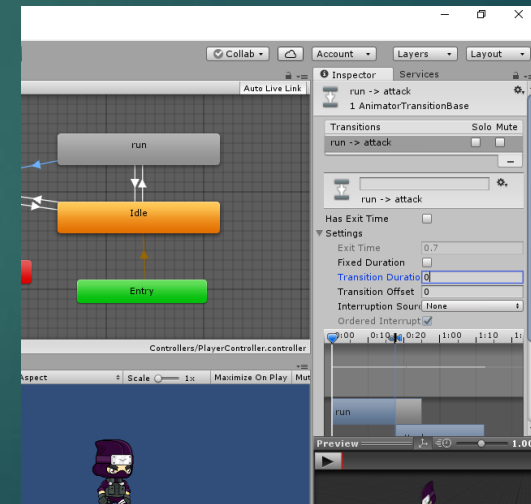
Ρυθμίσεις & transition του animation Attack

- ▶ Χρειάζεται προσοχή γιατί δεν μπορεί να τρέχει και να επιτίθεται ταυτόχρονα (δεν έχουμε τέτοιο animation). Σταματάει να τρέχει και επιτίθεται. Οπότε δε χρειάζεται να δημιουργήσουμε transition από το attack στο run, παρά μόνο transition από το attack στο idle και από εκεί στο run. Δημιουργούμε το transition από το attack στο idle.
- ▶ Σύνολο έχουμε δημιουργήσει 3 transition (run → attack, idle → attack, attack → idle)



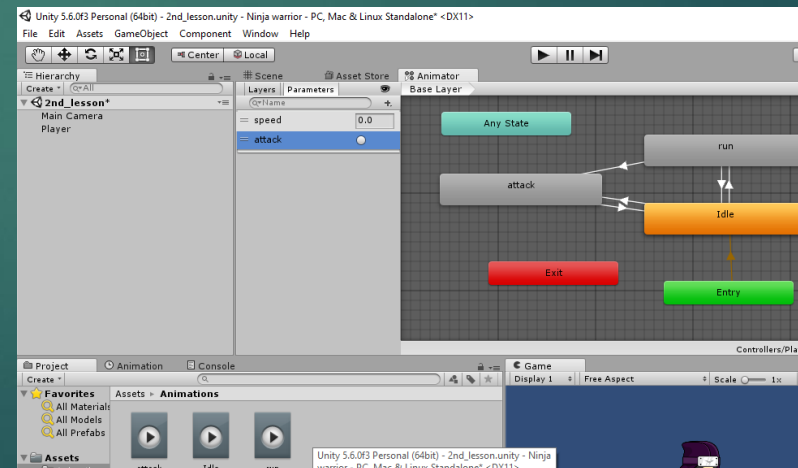
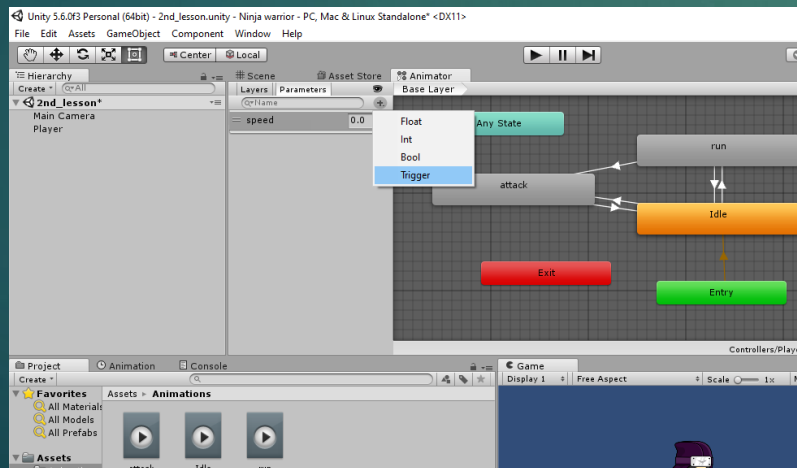
Ρυθμίσεις & transition του animation Attack

- ▶ Κάντε τις παρακάτω ρυθμίσεις στα transition που δημιουργήσατε ΕΚΤΟΣ ΑΠΟ ΤΟ TRANSITION ATTACK-IDLE! Αυτό το κάνουμε γιατί θέλουμε να επανέρχεται στην κατάσταση idle με το που τελειώνει η κατάσταση attack.
- ▶ Αποεπιλογή Has exit time
- ▶ Αποεπιλογή fixed duration &
- ▶ transition duration = 0



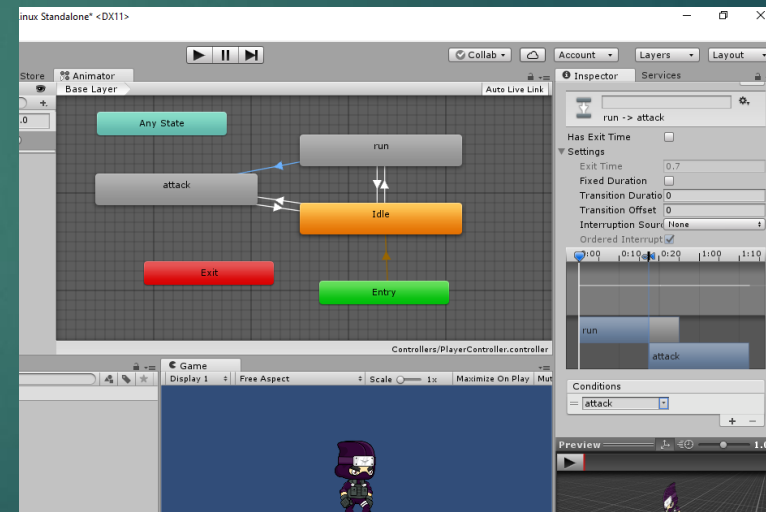
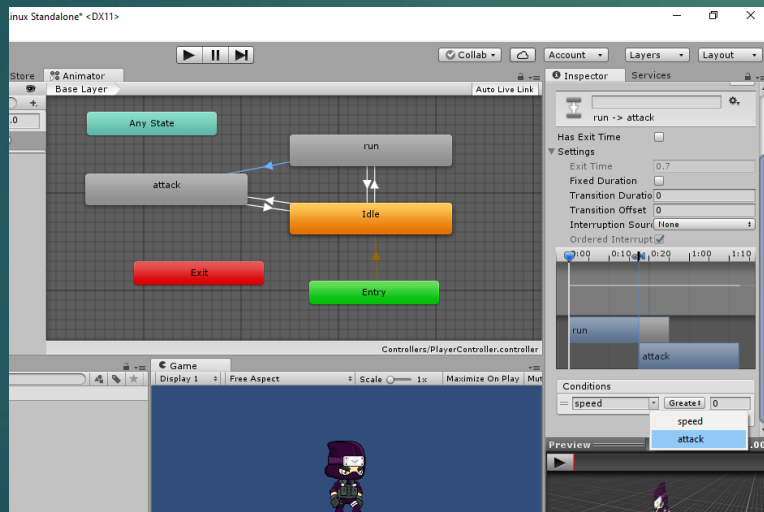
Conditions του animation Attack

- ▶ Τώρα θα ορίσουμε το condition του transition idle-attack το οποίο θα είναι trigger. Η διαφορά του trigger από τη bool parameter είναι ότι η bool μπορεί να είναι true-false, ενώ η trigger απλά «παιρνει ένα έναυσμα» και εκτελείται. Οπότε θα πρέπει να δημιουργήσουμε μια παράμετρο trigger και θα την ονομάσουμε attack.



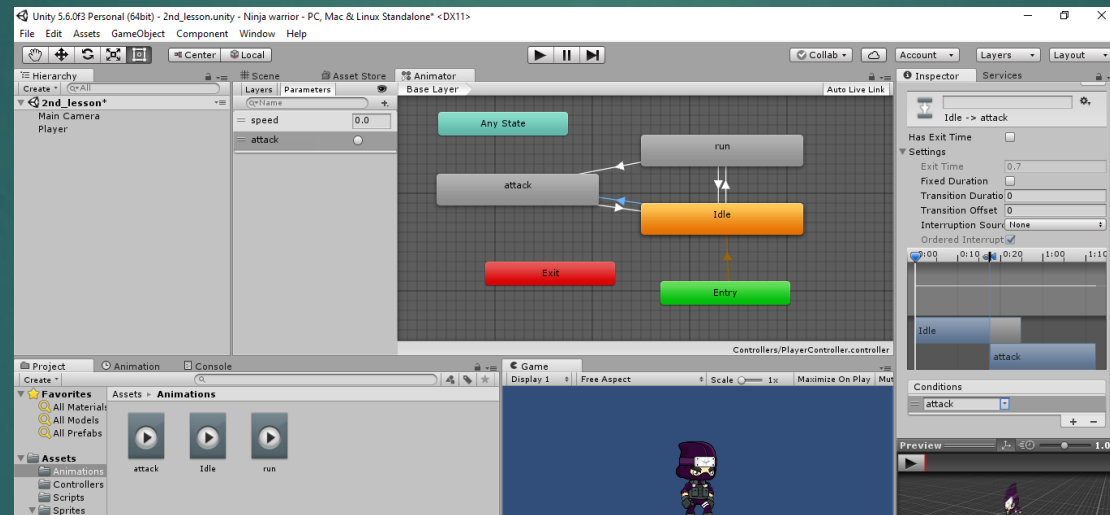
Conditions του animation Attack

- ▶ Επιλέγουμε το βελάκι του transition run→attack και προσθέτουμε το condition.
- ▶ Επιλέγουμε από τη λίστα των condition το attack



Conditions του animation Attack

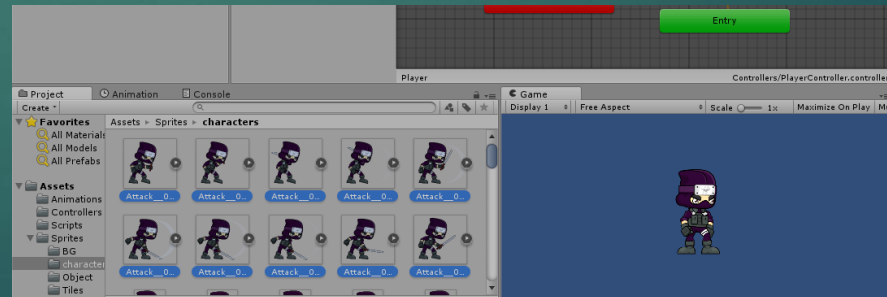
- ▶ Κάνουμε το ίδιο και για το Transition idle→attack



Δοκιμάζουμε τι έχουμε κάνει μέχρι τώρα

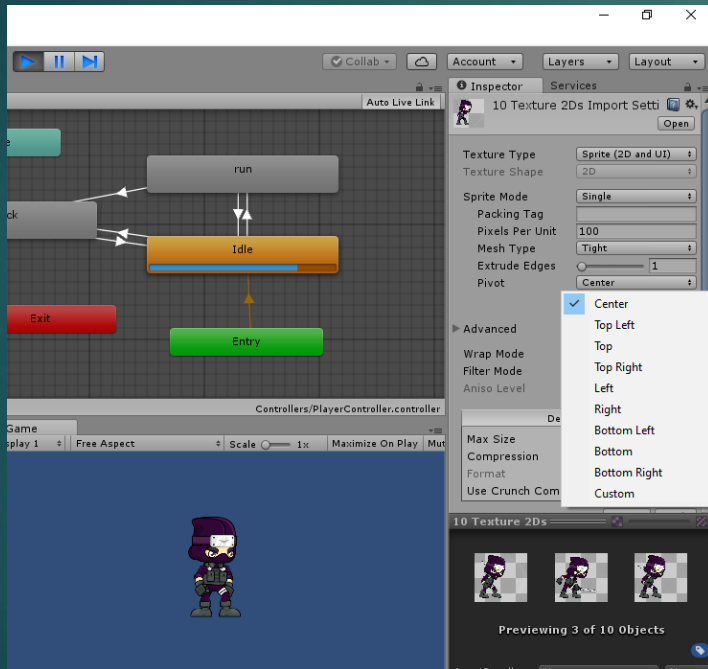
Ρυθμίσεις του animation Attack

- ▶ Όταν πατήσουμε πάνω στην παράμετρο attack θα δούμε το animation Attack το οποίο σταματάει μόλις ολοκληρωθεί. Θα παρατηρήσετε επίσης ότι ο Player μεταφέρεται λίγο προς τα αριστερά όταν εκτελείται το animation attack. Αυτό συμβαίνει γιατί οι εικόνες αυτού του animation είναι μεγαλύτερες σε μέγεθος. Θα πρέπει να το διορθώσουμε αυτό.
- ▶ Για να το διορθώσουμε μένουμε σε Play mode και επιλέγουμε από το φάκελο sprites τον φάκελο characters και από εκεί όλες τις εικόνες attack.



Ρυθμίσεις του animation Attack

Στον Inspector αλλάζετε το pivot από center σε custom. Αλλάζουμε γιατί το κέντρο δεν είναι ίδιο για όλες τις εικόνες, αφού είναι διαφορετικού μεγέθους.



Θα αλλάξουμε το X (με δοκιμές) για να δούμε ποιο είναι το κατάλληλο. Πατάμε το νούμερο, έπειτα apply και μετά πατάμε την παράμετρο attack για να δούμε απευθείας πως θα φαίνεται. Δοκιμάστε το 0.23. αν θέλετε δοκιμάστε να αλλάξετε και το Y.

Script Animation Attack

- ▶ Θα δημιουργήσουμε μια μεταβλητή

```
private bool attack;
```

- ▶ Και μια συνάρτηση

```
private void Handle Attacks()
```

```
{
```

```
if (attack)
```

```
{
```

```
myAnimator.SetTrigger("attack");
```

```
}
```

```
}
```

Script Animation Attack

- ▶ Δε δίνουμε έναυσμα (trigger) τώρα στη HandleAttack οπότε θα πρέπει να το προσθέσουμε. Για να το κάνουμε αυτό θα προσθέσουμε ακόμα μια συνάρτηση

```
private void HandleInput()  
{  
  If (Input.GetKeyDown(KeyCode.LeftShift))  
  {  
    myAnimator.SetTrigger("attack");  
  }  
}
```

Εδώ λέμε ότι αν πατηθεί το αριστερό Shift θα έχουμε επίθεση. Στην ουσία όταν πατηθεί θα γίνει true η bool μεταβλητή και θα τρέξει η HandleAttacks

Εδώ δίνεται το έναυσμα

Script Animation Attack

- ▶ Τώρα θα πρέπει να καλέσουμε τη `HandleInput` μέσα στην `Update`
- ▶ Ανάμεσα στη `Start` και τη `fixedupdate` θα προσθέσουμε μια `update`

```
Void Update()  
{  
  HandleInput();  
}
```

- ▶ Θα πρέπει επίσης να καλέσουμε κάπου τη `Handle Attack` οπότε προσθέτουμε στη `fixedUpdate`

```
HandleAttacks();
```


Δοκιμάζουμε

► Ολοκληρωμένος κώδικας

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

```
public class Player : MonoBehaviour {
```

```
    private Rigidbody2D myRigidbody;
    [SerializeField]
    private float movementSpeed;
    private bool attack;
    private bool facingRight;
    private Animator myAnimator;
```

```
    // Use this for initialization
```

```
    void Start ()
```

```
    {
        facingRight = true;
        myRigidbody = GetComponent<Rigidbody2D>();
        myAnimator = GetComponent<Animator>();
    }
```

```
    // Update is called once per frame
```

```
    void Update()
```

```
    {
        HandleInput();
    }
```

```
    void FixedUpdate ()
```

```
    {
        float horizontal = Input.GetAxis("Horizontal");
        HandleMovement(horizontal);
        Flip(horizontal);
        HandleAttacks();
    }
```

```
    private void HandleMovement(float horizontal)
```

```
    {
        myRigidbody.velocity = new Vector2(horizontal*movementSpeed,
        myRigidbody.velocity.y);
        myAnimator.SetFloat("speed", Mathf.Abs(horizontal));
    }
```

```
    private void HandleAttacks()
```

```
    {
        if(attack)
        {
            myAnimator.SetTrigger("attack");
        }
    }
```

```
    private void HandleInput()
```

```
    {
        if(Input.GetKeyDown(KeyCode.LeftShift))
        {
            attack = true;
        }
    }
```

```
    private void Flip(float horizontal)
```

```
    {
        if (horizontal > 0 && !facingRight || horizontal < 0 && facingRight)
        {
            facingRight = !facingRight;
            Vector3 theScale = transform.localScale;
            theScale.x *= -1;
            transform.localScale = theScale;
        }
    }
```

```
    }
}
```

```

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

public class Player : MonoBehaviour {

    private Rigidbody2D myRigidbody;
    [SerializeField]
    private float movementSpeed;
    private bool attack;
    private bool facingRight;
    private Animator myAnimator;

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

    // Update is called once per frame
    void Update()
    {
        HandleInput();
    }
    void FixedUpdate ()
    {
        float horizontal = Input.GetAxis("Horizontal");
        HandleMovement(horizontal);
        Flip(horizontal);
        HandleAttacks();
    }
    private void HandleMovement(float horizontal)
    {
        myRigidbody.velocity = new Vector2(horizontal*movementSpeed, myRigidbody.velocity.y);
        myAnimator.SetFloat("speed", Mathf.Abs(horizontal));
    }
    private void HandleAttacks()
    {
        if(attack)
        {
            myAnimator.SetTrigger("attack");
        }
    }
    private void HandleInput()
    {
        if(Input.GetKeyDown(KeyCode.LeftShift))
        {
            attack = true;
        }
    }

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

```

ΔΙΟΡΘΩΣΗ ΤΟΥ LOOP

- ▶ Αφαιρούμε από το animation Attack το Loop Time, αλλά το πρόβλημα δε διορθώνεται.
- ▶ Θα πρέπει να γυρίσουμε τη μεταβλητή attack με κάποιο τρόπο σε false, οπότε θα προσθέσουμε μια συνάρτηση

```
private void ResetValues()  
{  
    attack=false;  
}
```

Αυτή τη συνάρτηση θα τη χρησιμοποιούμε και για άλλα animation κτλ -πχ Jump

- ▶ Καλούμε αυτή τη συνάρτηση μέσα στην FixedUpdate

ΔΟΚΙΜΑΖΟΥΜΕ ΞΑΝΑ!

ΔΙΟΡΘΩΣΗ ΤΗΣ ΚΙΝΗΣΗΣ ΚΑΤ'Α ΤΗΝ ΕΠΙΘΕΣΗ

- ▶ Ένα ακόμα πρόβλημα που μπορούμε να εντοπίσουμε είναι ότι ο παίκτης δε σταματάει να κινείται ακόμα και όταν επιτίθεται. Θα πρέπει να σταματάει όταν επιτίθεται.

- ▶ Μέσα στη `HandleMovement` θα προσθέσουμε το παρακάτω

```
if(!this.myAnimator.GetCurrentAnimatorStateInfo(0).IsTag("Attack"))
```

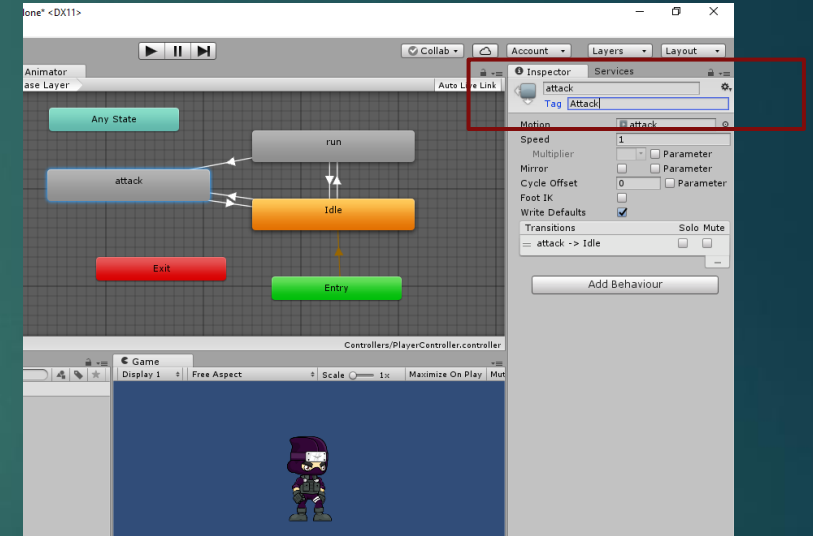
Και θα γίνει

```
if(!this.myAnimator.GetCurrentAnimatorStateInfo(0).IsTag("Attack"))  
  
    {  
  
        myRigidbody.velocity = new Vector2(horizontal * movementSpeed,  
myRigidbody.velocity.y);  
  
    }
```

Ελέγχει την τρέχουσα κατάσταση του animation. Ψάχνουμε ποιο animation παίζει και αν το tag ΔΕΝ είναι (!) Attack, εκτελείται η κίνηση

ΔΙΟΡΘΩΣΗ ΤΗΣ ΚΙΝΗΣΗΣ ΚΑΤ'Α ΤΗΝ ΕΠΙΘΕΣΗ

- ▶ Θα πρέπει να βάλουμε και Tag στο animation όμως αν θέλουμε να δουλέψει.
- ▶ Με επιλεγμένο το state Attack στον animator προσθέτουμε tag Attack
- ▶ Αν το δοκιμάσετε θα παρατηρήσετε ότι όσο επιτίθεται δε μπορεί να προχωρήσει και προχωράει μόνο όταν τελειώνει αυτή η κίνηση.



ΔΙΟΡΘΩΣΗ ΤΗΣ ΚΙΝΗΣΗΣ ΚΑΤ'Α ΤΗΝ ΕΠΙΘΕΣΗ

- ▶ Θα πρέπει τώρα να διορθώσουμε και το ότι όσο τρέχει μπορεί να επιτίθεται και φαίνεται σα να σέρνεται
- ▶ Αυτό θα το διορθώσουμε μέσα στη `HandleAttacks` προσθέτοντας το παρακάτω

```
myRigidbody.velocity=Vector2.zero;
```

- ▶ Δοκιμάστε το: Θα πρέπει να σταματά πριν κάνει την επίθεση

ΔΙΟΡΘΩΣΗ ΤΗΣ ΚΙΝΗΣΗΣ ΚΑΤ'Α ΤΗΝ ΕΠΙΘΕΣΗ

- ▶ Θα πρέπει να κάνουμε και μια τελευταία διόρθωση που αφορά να μην μπορούμε να ξεκινήσουμε μια νέα επίθεση αν δεν έχει τελειώσει η προηγούμενη.
- ▶ Για να το κάνουμε αυτό θα προσθέσουμε στο if της HandleAttacks

```
!this.myAnimator.GetCurrentAnimatorStateInfo(0).IsTag("Attack"))
```

Και θα γίνει

```
if(attack &&  
!this.myAnimator.GetCurrentAnimatorStateInfo(0).IsTag("Attack"))
```

ΔΟΚΙΜΑΖΟΥΜΕ ΑΝ ΔΟΥΛΕΥΕΙ