



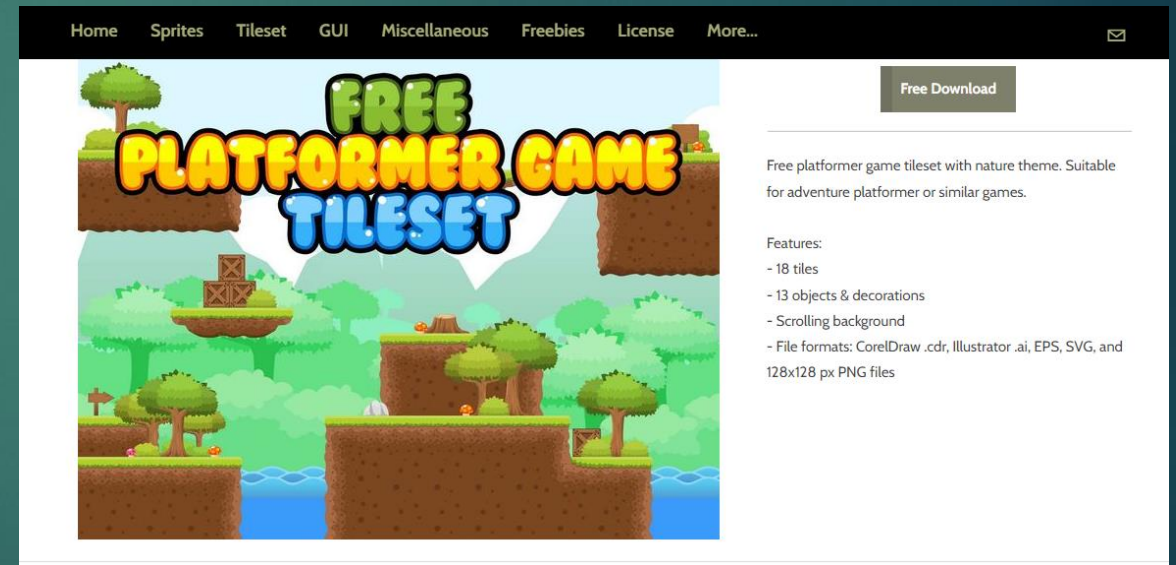
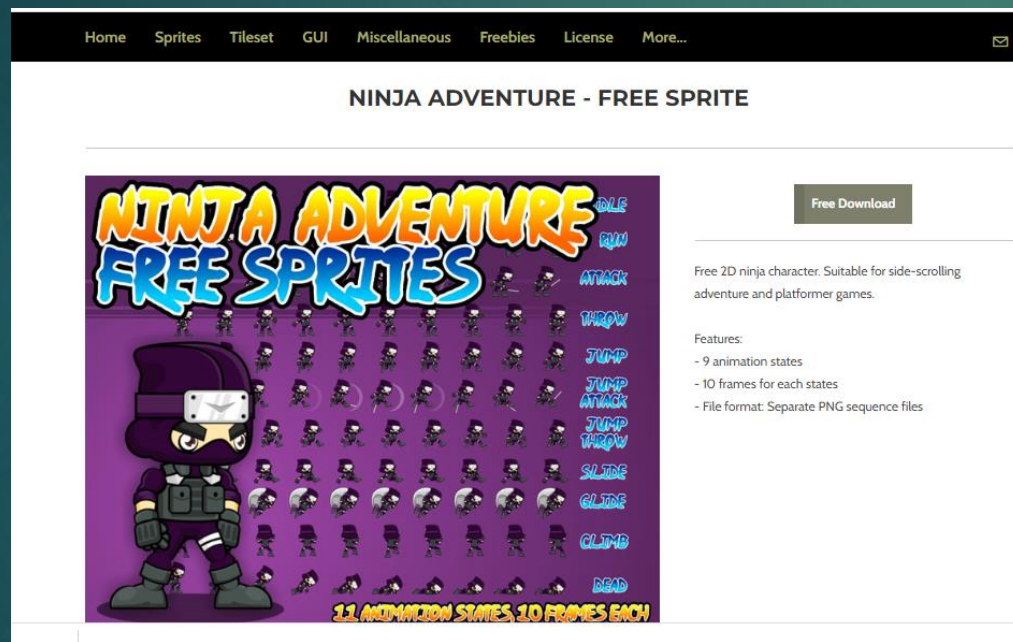
Δημιουργία 2D Platform

ΠΡΟΕΤΟΙΜΑΣΙΑ ΚΑΙ ΒΑΣΙΚΗ ΚΙΝΗΣΗ

ΣΤΕΛΛΑ ΚΑΙΝΟΥΡΓΙΑΚΗ
2022-2023

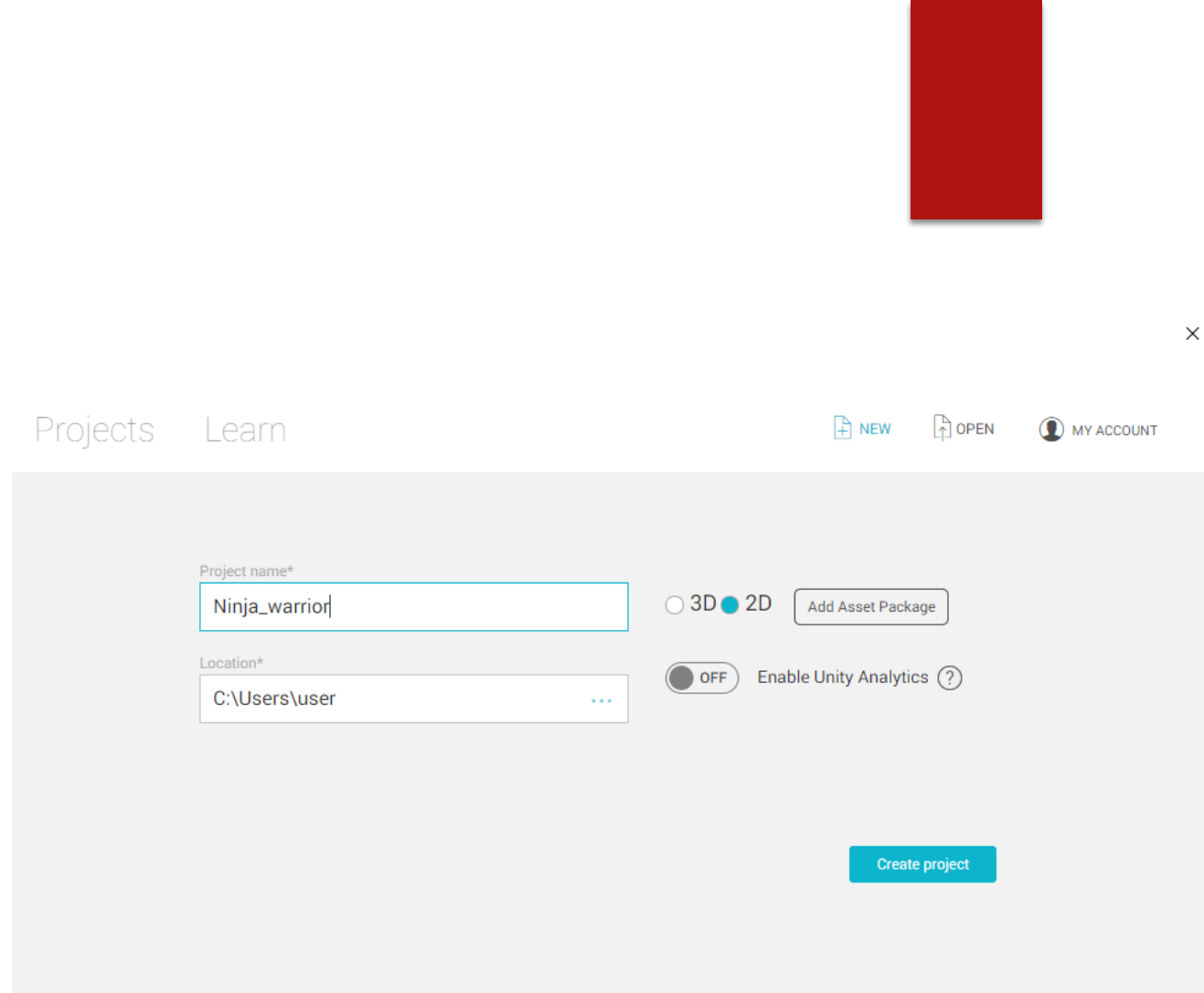
Προετοιμασία του project

- ▶ Μπείτε στο www.gameart2d.com και κατεβάστε από την καρτέλα freebies, το πακέτο Ninja Adventure free sprites και το tileset.



Προετοιμασία του project

Δημιουργούμε ένα νέο project και το ονομάζουμε Ninja warrior. Το project μας είναι 2D.



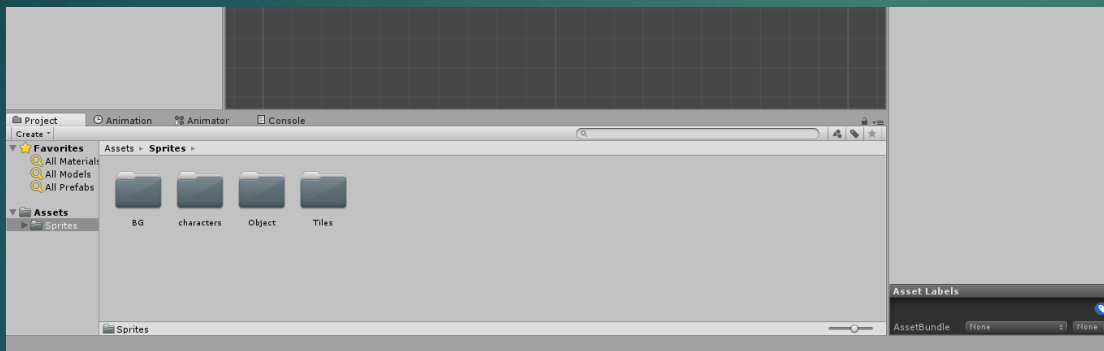
Προετοιμασία του project

- ▶ Αφού έχουμε κατεβάσει τα αρχεία στον υπολογιστή και τα έχουμε αποσυμπιέσει, θα πρέπει να εισάγουμε τους φακέλους στο project μας.
- ▶ Δημιουργούμε στο project έναν φάκελο sprites και μέσα τοποθετούμε το πακέτο Ninja και το πακέτο που βρίσκεται στο φάκελο png του freetileset.

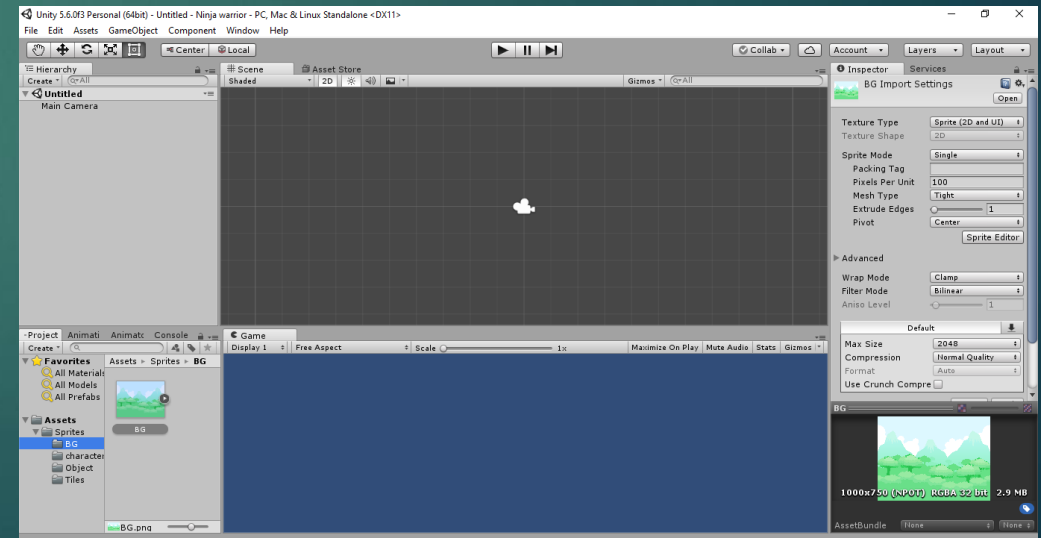


Προετοιμασία του project

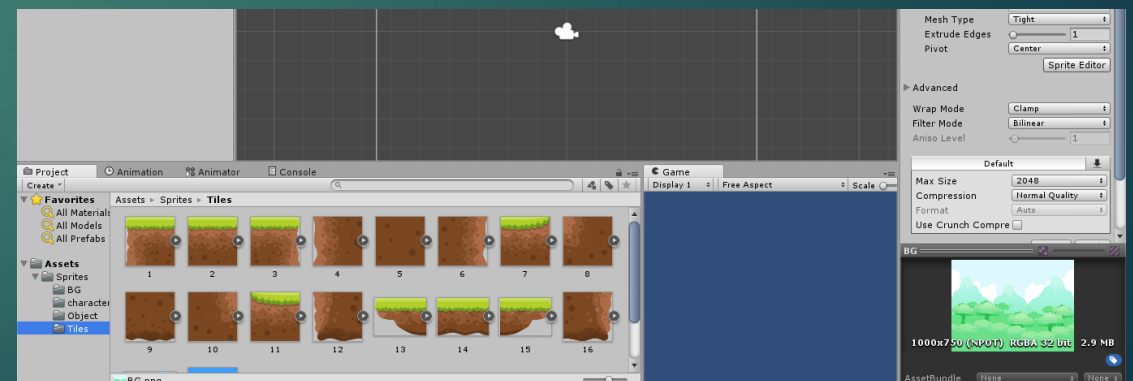
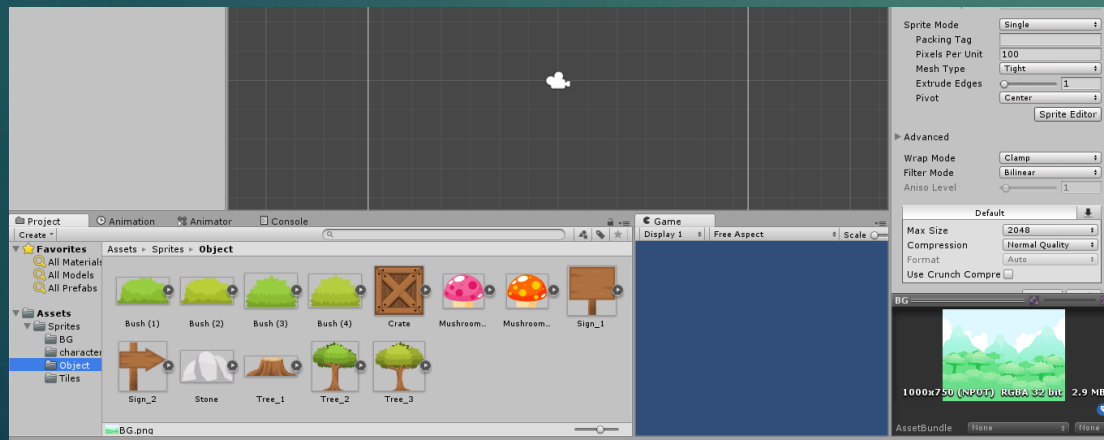
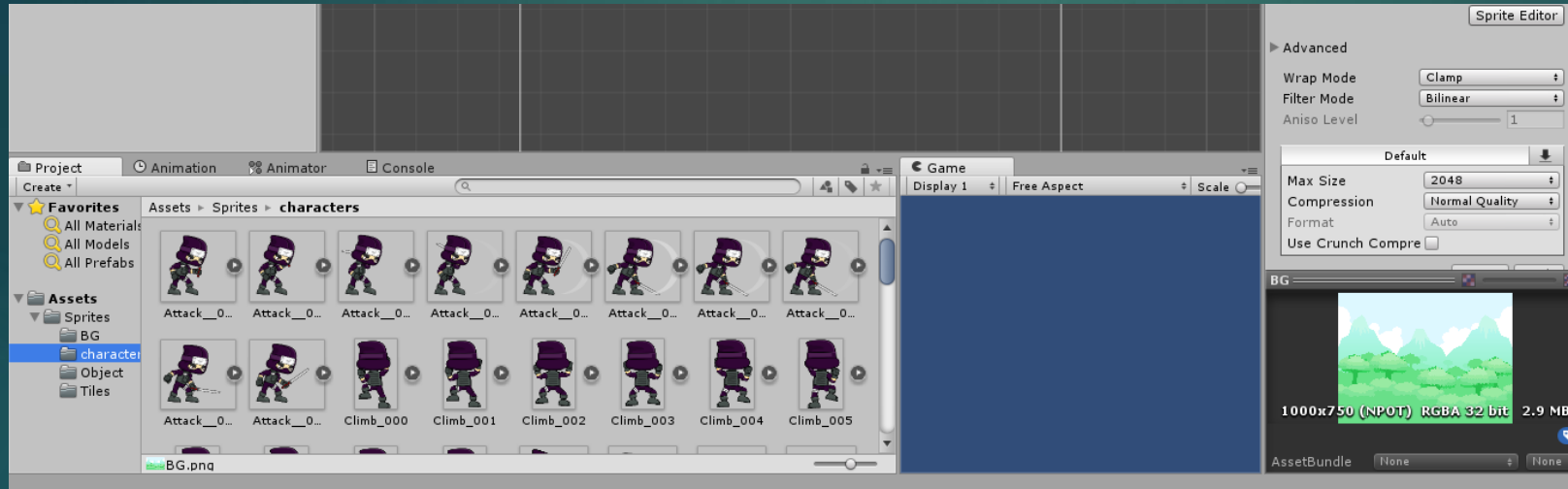
- ▶ Ο φάκελος sprites θα πρέπει να περιέχει τα παρακάτω (έχω μετονομάσει το φάκελο ninja σε characters)



Με κλικ στον κάθε φάκελο θα δείτε στα δεξιά τα περιεχόμενά του.

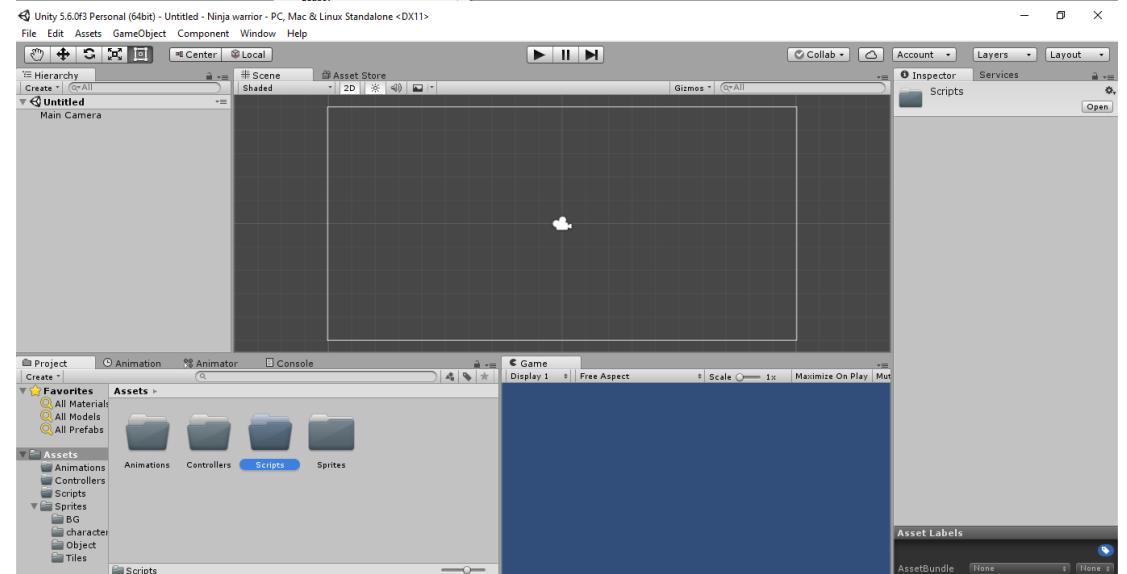
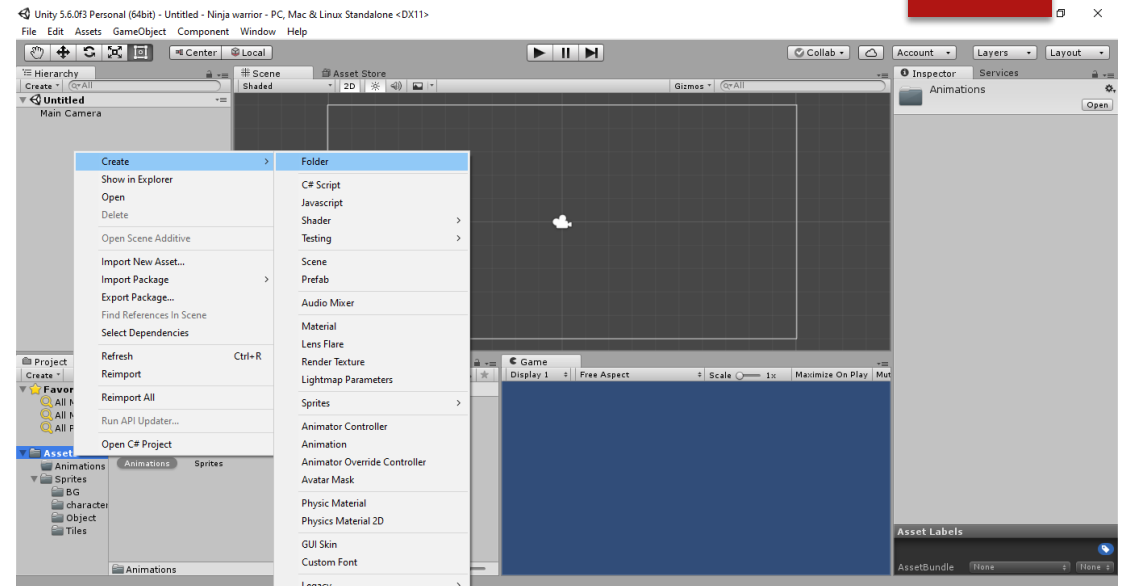


Προετοιμασία του project – περιεχόμενο φακέλων



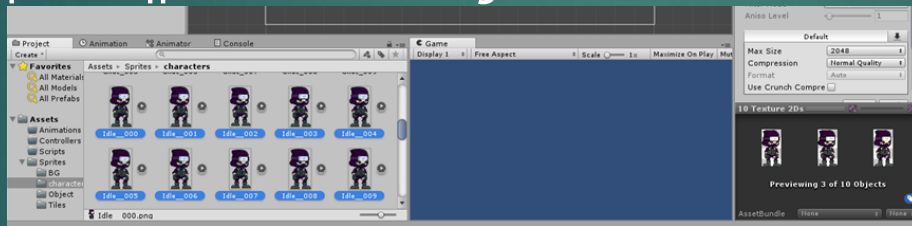
Προετοιμασία του project

- ▶ Δημιουργούμε 3 νέους φακέλους:
- ▶ animation
- ▶ controllers
- ▶ scripts



Δημιουργία animation

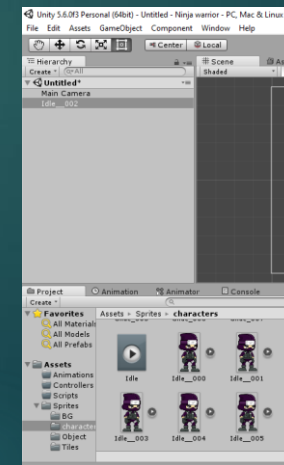
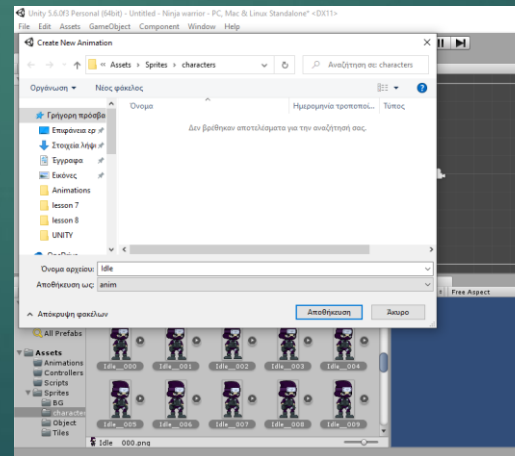
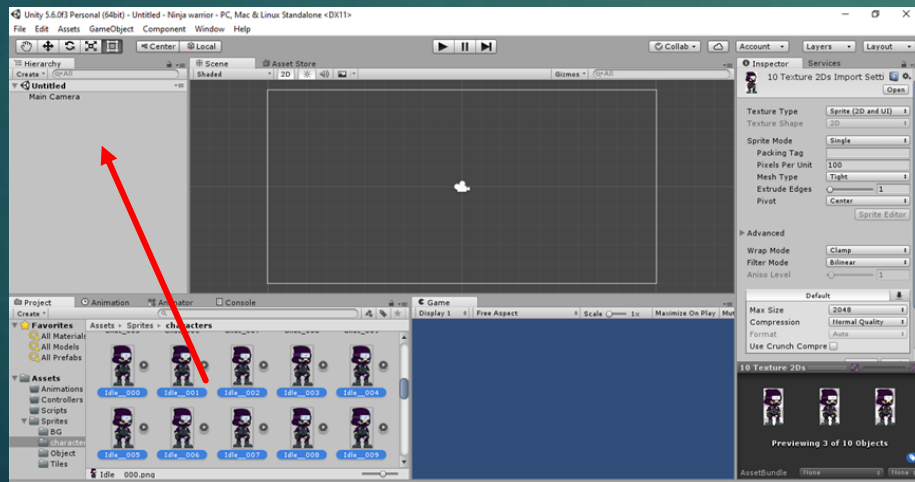
- ▶ Στο φάκελό sprites-characters θα βρείτε όλα τα στιγμιότυπα που μπορούμε να χρησιμοποιήσουμε για το animation. Για παράδειγμα οι εικόνες με όνομα idle_xx είναι αυτές που δημιουργούν το animation της όρθιας στάσης για τον παίκτη μας. Παρατηρείστε ότι ξεκινάνε από το 000 και φτάνουν μέχρι το 009.



- ▶ Ένας εύκολος τρόπος για να δημιουργήσουμε animation είναι ο παρακάτω:
- ▶ Επιλέγουμε όλα τα αρχεία που θέλουμε να συμπεριληφθούν στο animation, στην περίπτωση μας τα idle000 μέχρι idle 009

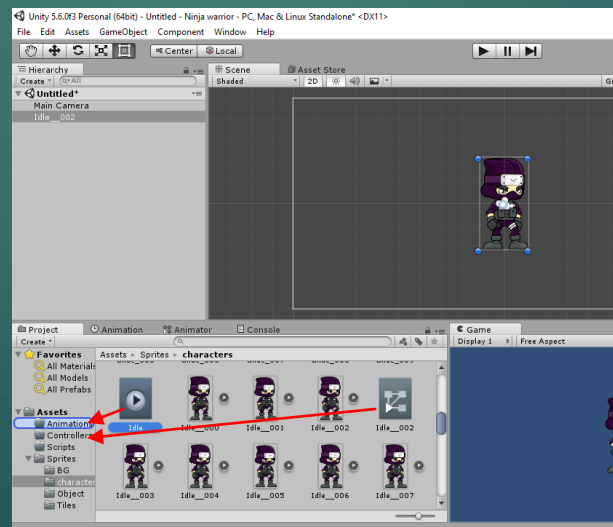
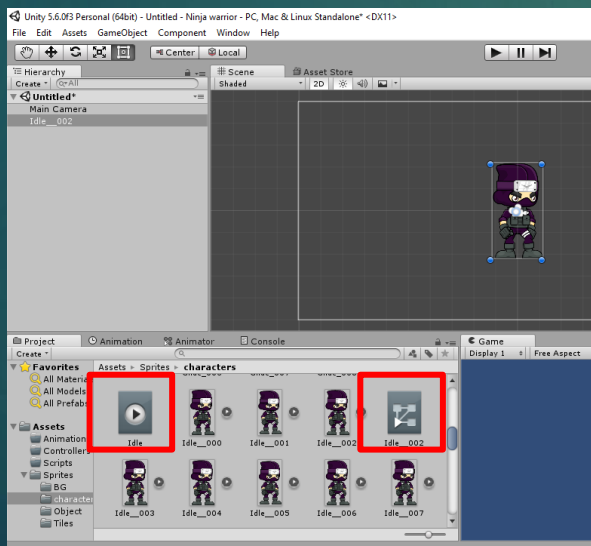
Δημιουργία animation

- ▶ Σέρνουμε όλα τα αρχεία στην ιεραρχία και θα δούμε ότι εμφανίζεται ένα παράθυρο δημιουργίας animation. Ονομάζουμε το animation Idle και πατάμε Αποθήκευση.
- ▶ Παρατηρούμε ότι στην ιεραρχία δεν τοποθετούνται 9 αρχεία αλλά 1 στοιχείο με όνομα ίδιο με το όνομα του τελευταίου από τα αρχεία που επιλέξαμε (Idle_009).



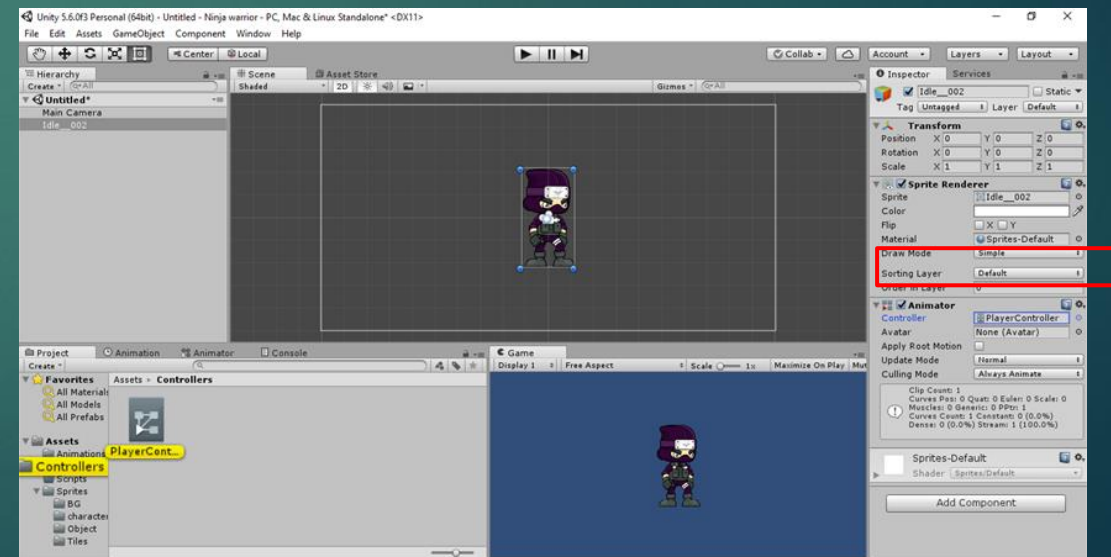
Δημιουργία animation

- ▶ Θα δούμε επίσης ότι στο φάκελο Characters δημιουργήθηκε ένα animation με όνομα idle_000 καθώς και ένας controller.
- ▶ Για να είναι «τακτοποιημένα» τα αρχεία μας, θα μεταφέρουμε το animation idle στο φάκελο animations και τον Controller στο φάκελο Controllers



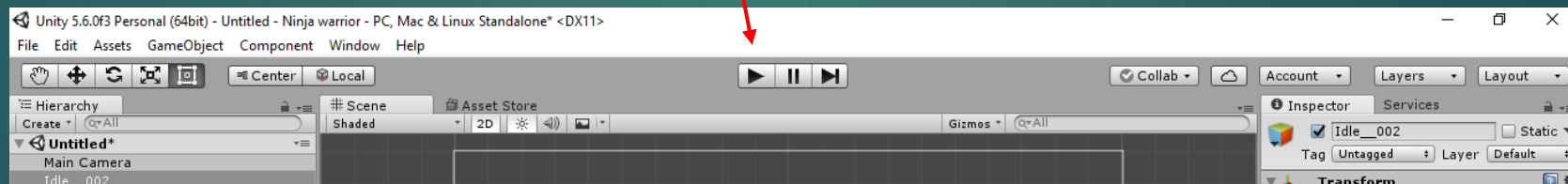
Δημιουργία animation

- ▶ Μετονομάζουμε τον Controller σε PlayerController. Μετονομάζουμε επίσης το idle που βρίσκεται στην ιεραρχία σε Player.
- ▶ Θα δούμε τώρα ότι αν επιλέξουμε το idle που υπάρχει στην ιεραρχία, έχει ήδη προτεθεί το component animator καθώς και ότι ο animator controller έχει αντιστοιχηθεί με τον PlayerController. Αν κάνουμε κλικ πάνω στον animator controller θα δούμε ότι έχει αναφορά στον controller μας (γίνονται κίτρινα). Οποιαδήποτε αλλαγή θέλουμε να κάνουμε με τα animation του παίκτη, πρέπει να γίνεται μέσα από αυτόν τον controller.



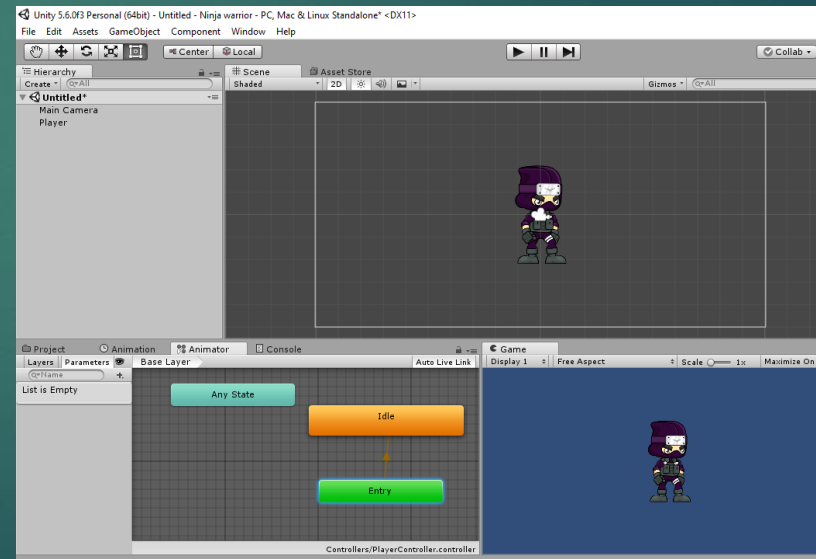
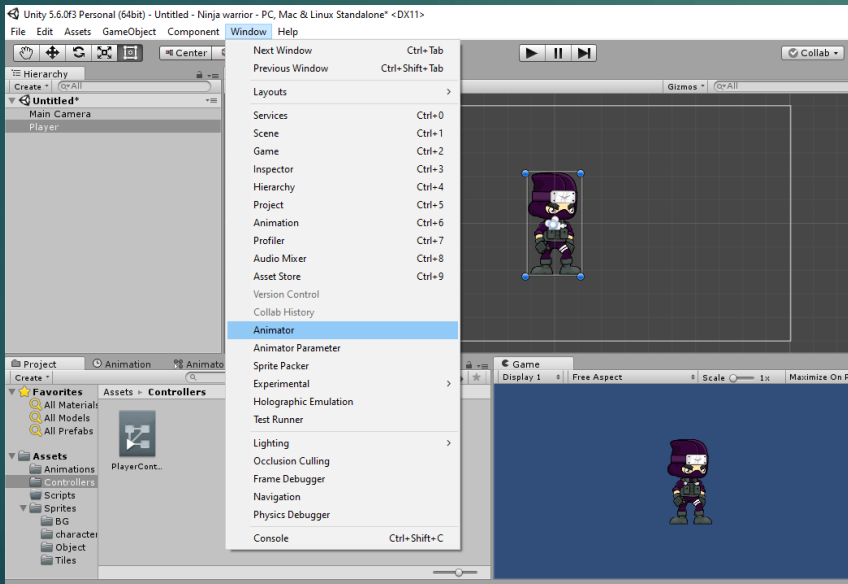
Δημιουργία animation – πρώτη δοκιμή

- ▶ Αν πατήσετε το play mode θα δείτε ότι ο παίκτης έχει κίνηση
- ▶ ΠΡΟΣΟΧΗ! Όταν είμαστε σε Play mode, οποιαδήποτε αλλαγή και αν κάνουμε στο project μας, ΔΕΝ αποθηκεύεται και «χάνεται» μόλις βγούμε από play mode.
- ▶ Πατάμε το Play mode για να δούμε το παιχνίδι και βγαίνουμε αμέσως από το Play Mode.



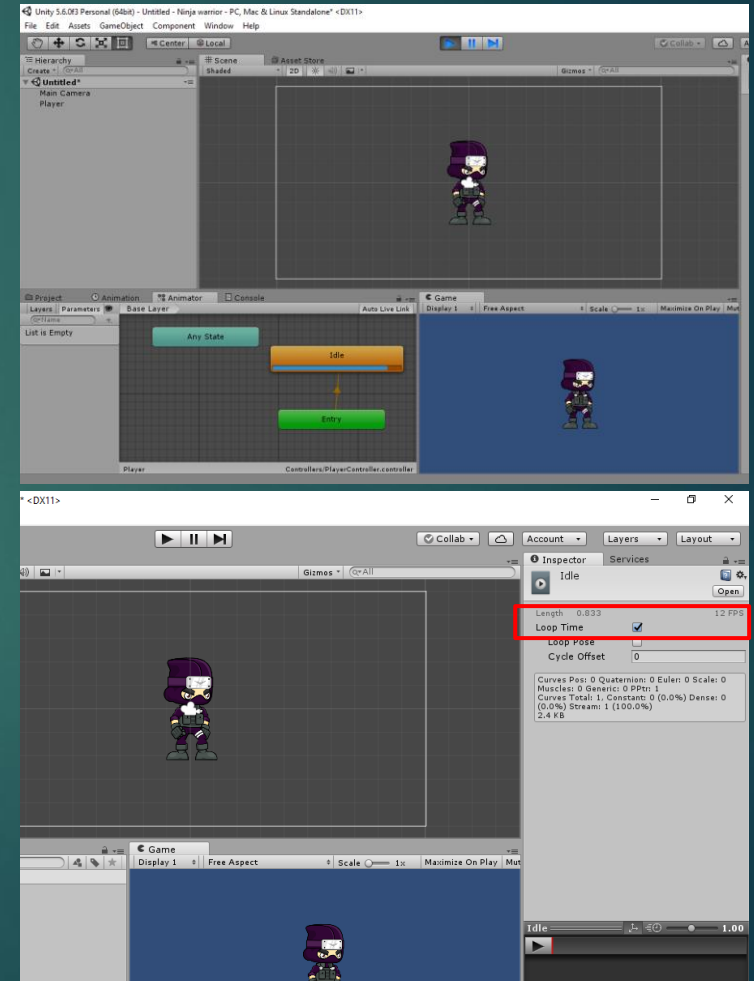
Δημιουργία animation – animator

- ▶ Για να δούμε τι συμβαίνει μέσα στον animator controller θα ανοίξουμε το παράθυρο Animator. Αν δεν υπάρχει στο περιβάλλον εργασία θα το βρούμε από το menu window → animator



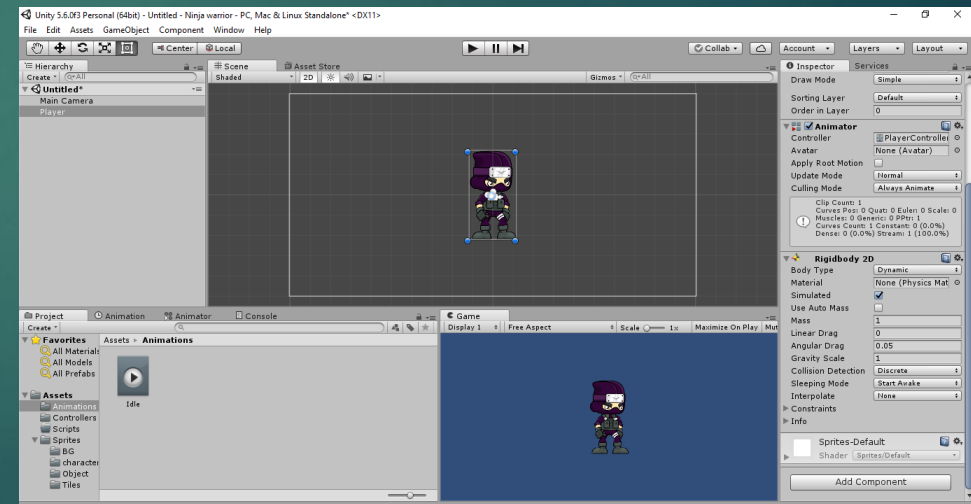
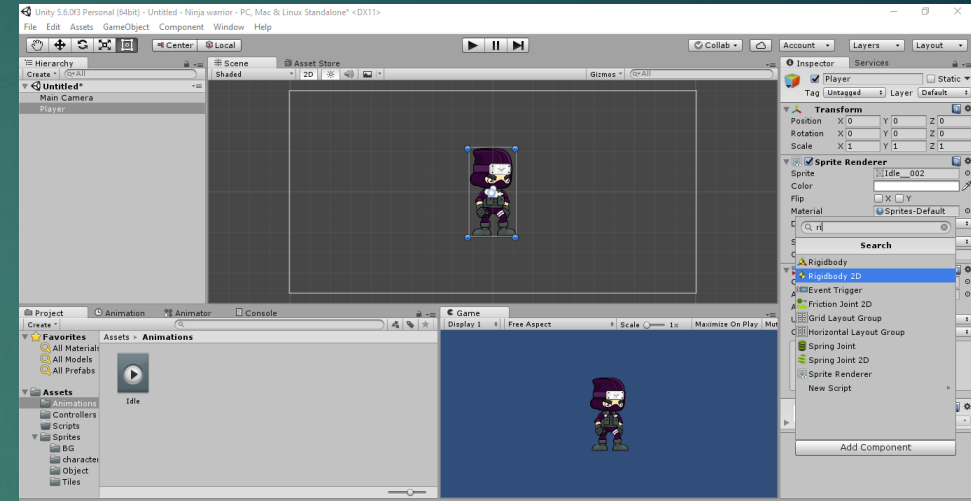
Δημιουργία animation – animator

- ▶ Το animation που είναι πορτοκαλί είναι αυτό που είναι default. Αν πατήσετε play mode θα δείτε ότι αυτό το animation παίζει συνέχεια (εμφανίζεται μπάρα προόδου στο κάτω μέρος)
- ▶ Το animation αυτό θα παίζει συνέχεια γιατί είναι επιλεγμένο το loop time (βρίσκεται στις ιδιότητες του animation στο inspector).



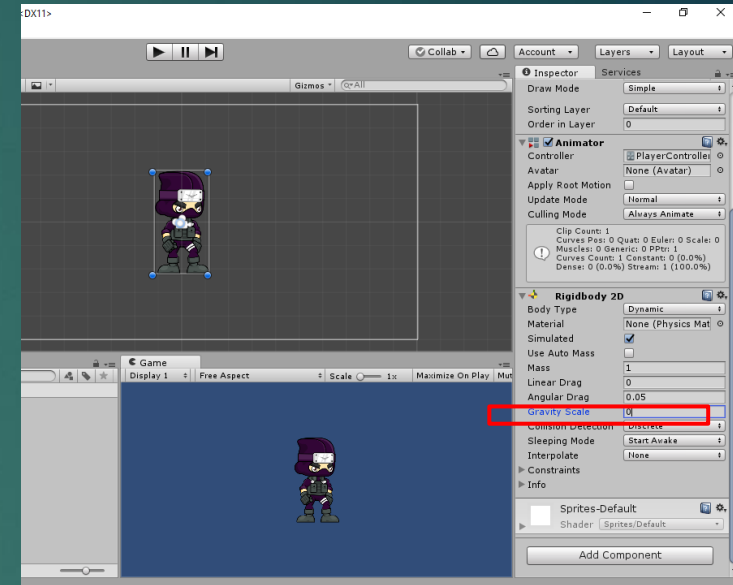
Δίνουμε κίνηση στον Player

Για να μπορέσουμε να προσδώσουμε κίνηση στον Player θα χρειαστεί να προσθέσουμε 2 components. Το πρώτο είναι το Rigidbody, που στην ουσία είναι αυτό που επικοινωνεί με τη φυσική στη Unity. Επιλέγουμε τον Player και στον inspector επιλέγουμε Add Component και ψάχνουμε το rigidBody 2D.



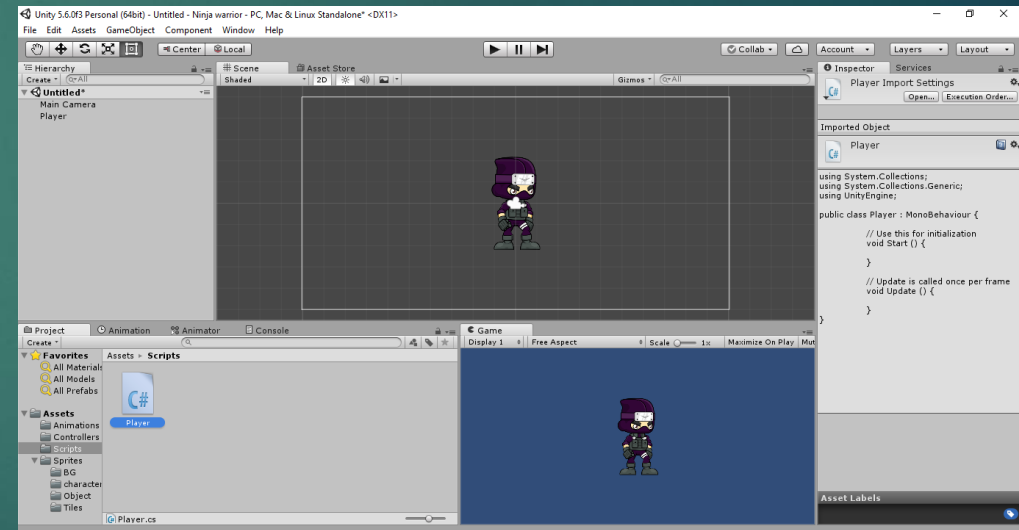
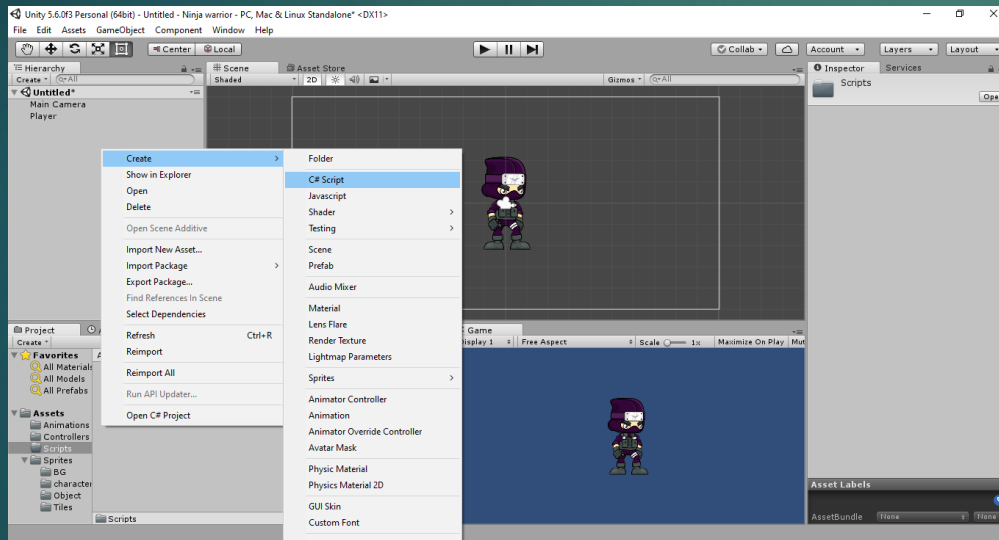
Δίνουμε κίνηση στον Player

- ▶ Αν πατήσετε Play Mode θα δείτε ότι ο παίκτης θα αρχίσει να πέφτει. Για να μπορέσουμε να δουλέψουμε το παιχνίδι μας χωρίς να πέφτει ο Player κάθε φορά που θα μπαίνουμε σε Play Mode, θα αλλάξουμε το Gravity Scale (πεδίο στο Rigidbody 2D) σε 0.



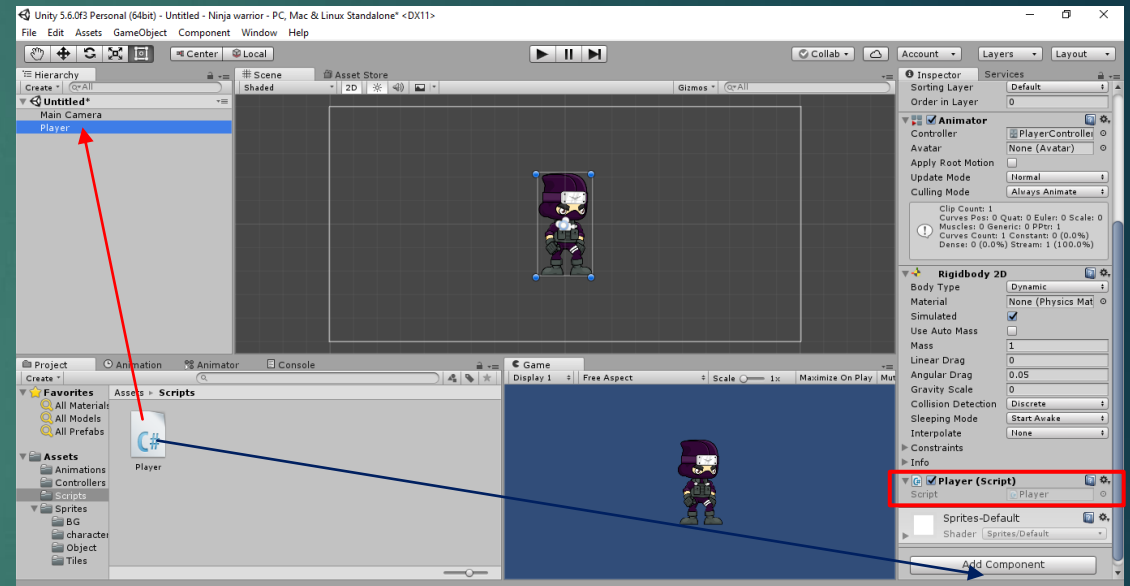
Δίνουμε κίνηση στον Player

- ▶ Για να μπορέσουμε να δώσουμε κίνηση θα γράψουμε script. Πηγαίνουμε στο φάκελο scripts και δημιουργούμε ένα νέο C# script που θα ονομάσουμε Player.



Δίνουμε κίνηση στον Player

- ▶ Για να κάνουμε attach το script στον Player υπάρχουν 2 τρόποι: είτε να σύρουμε το script στο πάνελ inspector (κάτω από το Add Component) είτε να σύρουμε το script στον Player που βρίσκεται στην ιεραρχία. Και στις 2 περιπτώσεις θα εμφανιστεί στο πάνελ inspector ένα νέο component, το Player (Script).



Δίνουμε κίνηση στον Player – ανάλυση κώδικα

- ▶ Η κίνηση του παίκτη θα γίνει μέσα από το component `rigidbody` οπότε θα πρέπει μέσα στο script να δημιουργήσουμε μια αναφορά στο `rigidbody`

```
private Rigidbody2D myRigidbody; // δημιουργούμε μια κενή μεταβλητή
```

- ▶ Για να κάνουμε τη συσχέτιση / αναφορά γράφουμε
`myRigidbody=GetComponent<Rigidbody2D>();`

Δίνουμε κίνηση στον Player – ανάλυση κώδικα

- ▶ Θα πρέπει επίσης να δημιουργήσουμε μια συνάρτηση έτσι ώστε να κινείται ο παίκτης

```
private void HandleMovement() // θα διαχειρίζεται την κίνηση
```

```
{
```

```
myRigidbody.velocity = Vector2.left; // θα πρέπει να εφαρμόσουμε  
μια δύναμη (velocity) έτσι ώστε να κινείται τριγύρω – το vector2.left  
είναι ένα διάνυσμα με τιμές X=-1 και Y=0
```

```
}
```

- ▶ Τώρα θα πρέπει να καλέσουμε τη συνάρτηση HandleMovement στην Update (αλλιώς δεν έχει νόημα!)

```
HandleMovement();
```

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

public class Player : MonoBehaviour {

    private Rigidbody2D myRigidbody;

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

    // Update is called once per frame
    void Update ()
    {
        HandleMovement();
    }
    private void HandleMovement()
    {
        myRigidbody.velocity = Vector2.left;
    }
}
```

Δίνουμε κίνηση
στον Player –
ολοκληρωμένος
κώδικας

Δίνουμε κίνηση στον Player

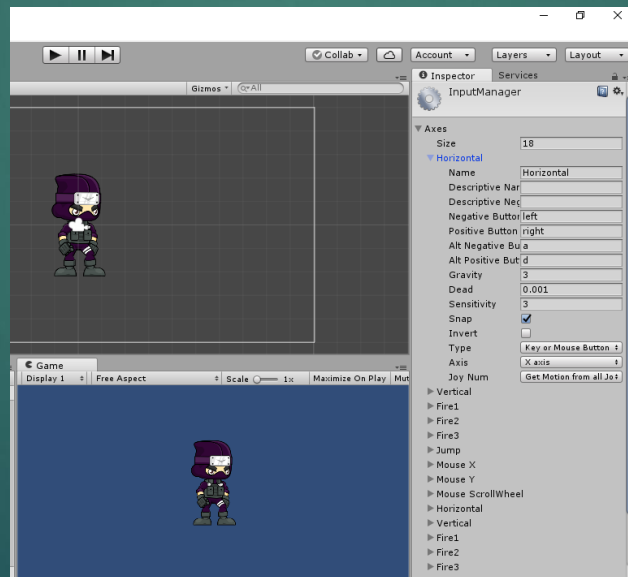
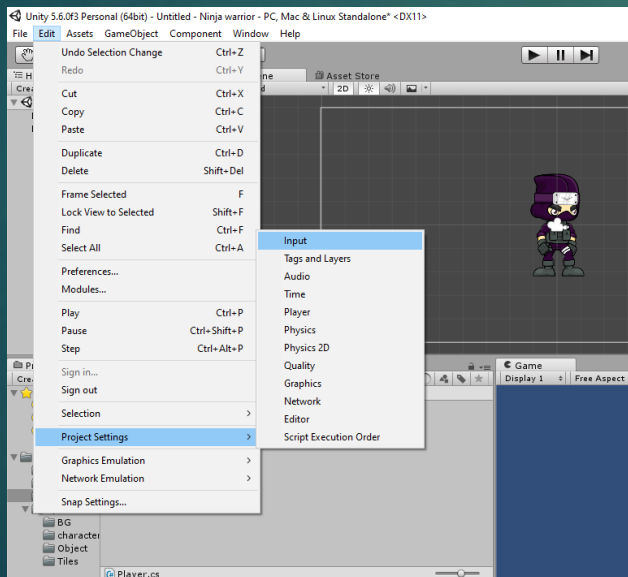
Τι συμβαίνει αν πατήσουμε το Play Mode;;

Ο παίκτης θα πρέπει να κινείται προς τα αριστερά μόνος του.

Σε αυτό το σημείο θα ξεκινήσουμε να προσθέτουμε λειτουργικότητες στην κίνηση του παίκτη έτσι ώστε να κινείται μετά από δική μας εντολή

Δίνουμε κίνηση στον Player

- ▶ Στη Unity υπάρχει η κλάση Input η οποία ορίζει εντολές εισόδου. Η κλάση αυτή έχει συναρτήσεις, μία από τις οποίες είναι η Horizontal και αφορά εντολές εισόδου για την οριζόντια κίνηση. Μπορείτε να δείτε τις επιλογές εισόδου από το μενού Edit – Project Settings – Input



Εδώ μπορούμε να δούμε τις παραμέτρους που επηρεάζουν τον οριζόντιο άξονα. Βλέπουμε ότι την τιμή στον οριζόντιο άξονα την επηρεάζουν τα βελάκια (δεξιά και αριστερά) αλλά και τα πλήκτρα a & d.

Δίνουμε κίνηση στον Player

- ▶ Θα προσθέσουμε την horizontal axis μέσα στην update

```
float horizontal = Input.GetAxis("Horizontal")
```

// σε αυτή τη γραμμή λοιπόν η τιμή από τον άξονα X αποθηκεύεται στη μεταβλητή horizontal.

- ▶ Θα χρησιμοποιήσουμε αυτήν την τιμή στη handleMovement σαν παράμετρο

```
HandleMovement(horizontal);
```

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

```
myRigidbody.velocity = new Vector2(horizontal,  
myRigidbody.velocity.y); // ζητάει την τιμή X που είναι  
αποθηκευμένο στην horizontal, κρατάμε ίδιο το y
```



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

public class Player : MonoBehaviour {

    private Rigidbody2D myRigidbody;

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

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

ΔΟΚΙΜΑΖΟΥΜΕ
ΤΟ PLAY MODE

ολοκληρωμένος
κώδικας