



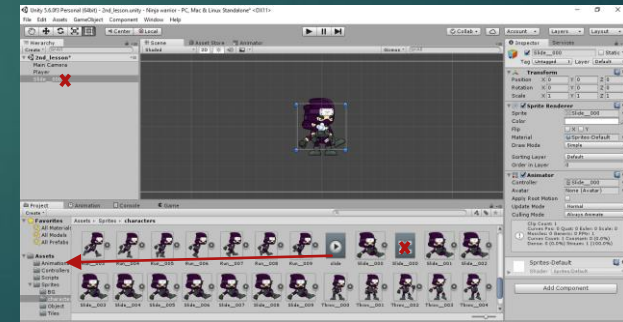
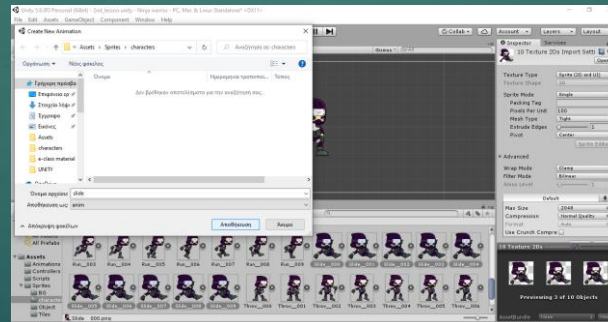
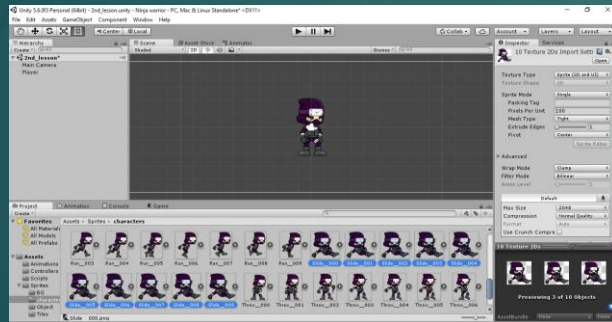
Δημιουργία 2D Platform

ΑΛΛΑΓΕΣ & ΔΙΟΡΘΩΣΕΙΣ ΣΤΗΝ ΚΙΝΗΣΗ ΤΟΥ ΠΑΙΚΤΗ – ΝΕΑ ANIMATION

ΣΤΕΛΛΑ ΚΑΙΝΟΥΡΓΙΑΚΗ
2022-2023

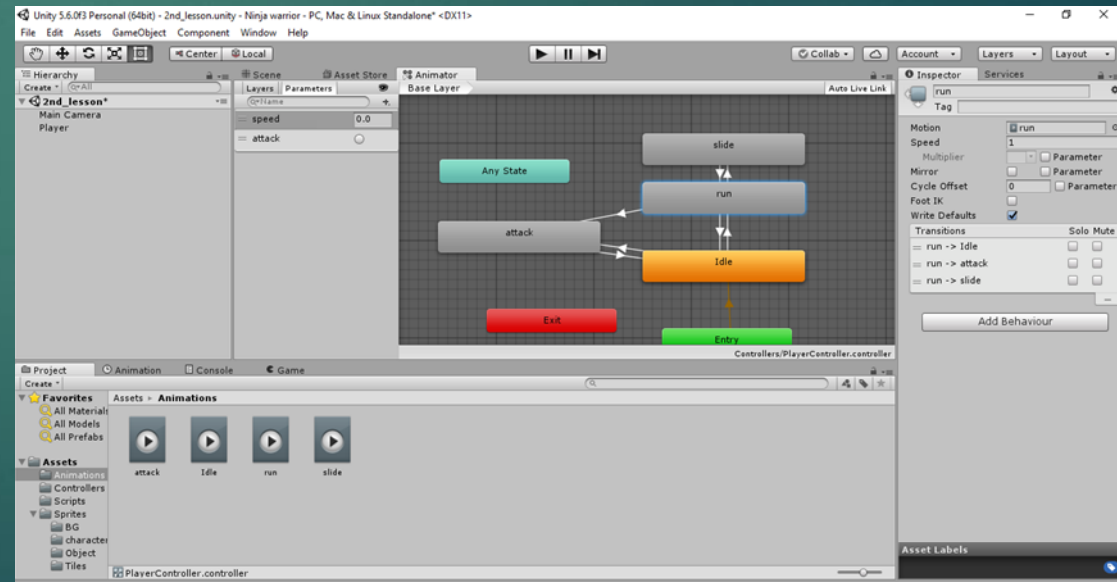
Προσθήκη animation slide

- ▶ Με τον ίδιο τρόπο που έχουμε κάνει όλα τα προηγούμενα animation θα δημιουργήσουμε ένα animation slide. Ο παίκτης θα μπορεί να κάνει slide μόνο όσο τρέχει.
- ▶ Επιλέγουμε τα sprites slide και τα σέρνουμε στην ιεραρχία. Ονομάζουμε το animation slide, διαγράφουμε το νέο στοιχείο που δημιουργείται στην ιεραρχία, διαγράφουμε τον controller και μετακινούμε το animation slide στο φάκελο με τα animation.



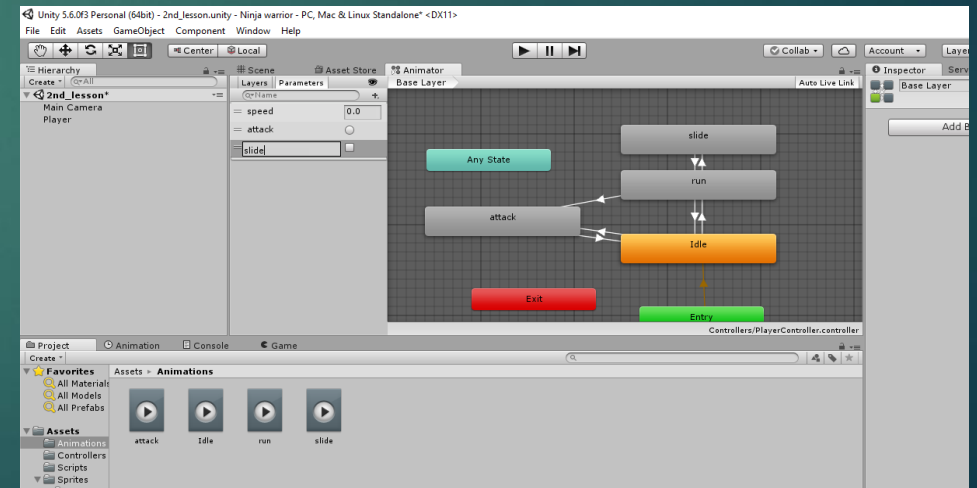
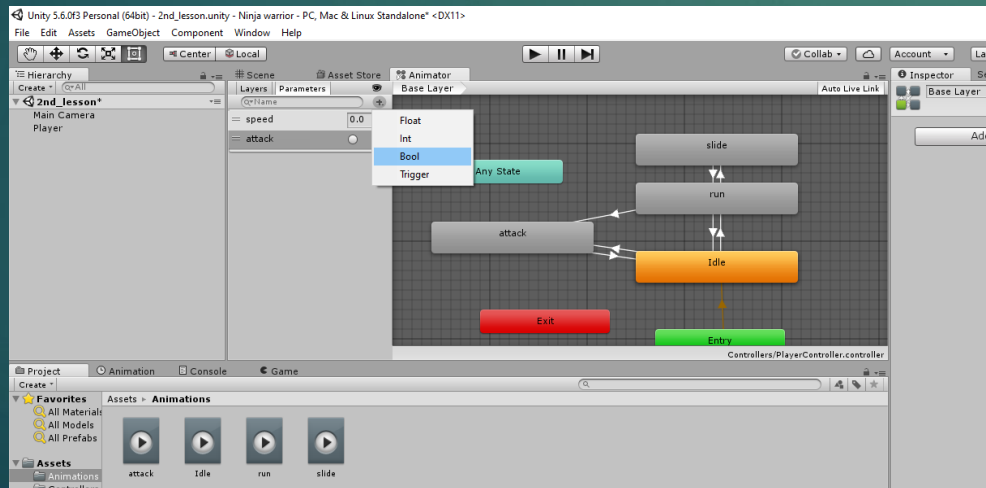
Προσθήκη στον animator

- ▶ Σέρνουμε το νέο animation στον animator, κοντά στο animation run.
- ▶ Τώρα θα δημιουργήσουμε transition από το slide στο run και το αντίστροφο.



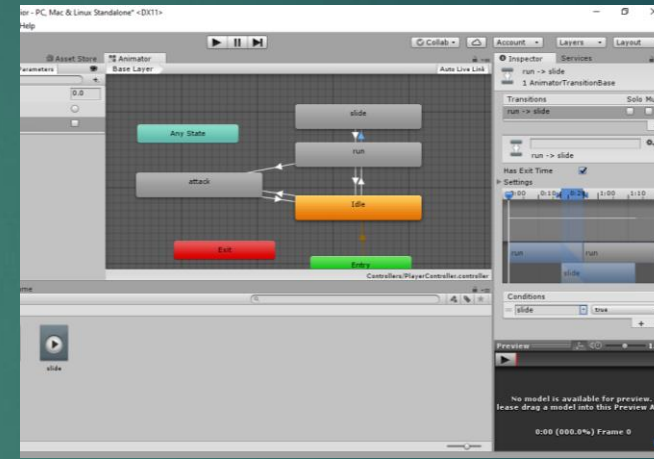
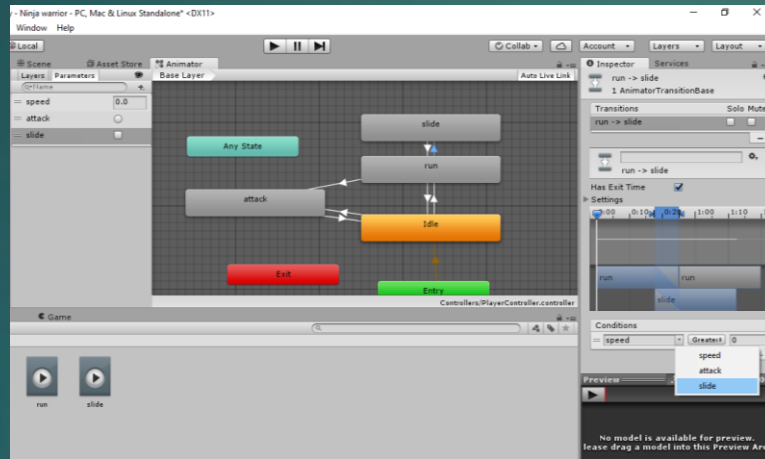
Ρυθμίσεις transition

- ▶ Δε χρειάζεται να πειράξουμε κάτι στο transition slide-run γιατί στην ουσία το slide θα ξεκινάει με το πάτημά ενός πλήκτρου και θα επιστρέφει στο run. Αυτό στο οποίο θα πρέπει να προσθέσουμε λειτουργικότητα είναι το transition run-slide. Θα χρησιμοποιήσουμε μια παράμετρο Boolean που θα ονομάσουμε slide (αν είναι true θα ξεκινήσει να κάνει slide ο Player).



Ρυθμίσεις transition

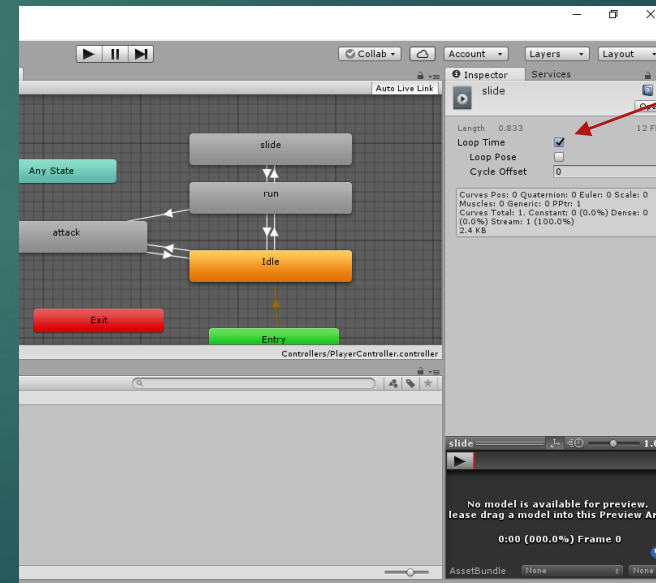
- ▶ Με επιλεγμένο το transition run→slide, θα προσθέσουμε condition με το parameter slide, έτσι ώστε να ενεργοποιείται όταν το slide είναι true.



- ▶ Κάνουμε επιπλέον τις παρακάτω ρυθμίσεις:
- ▶ Αποεπιλογή Has Exit time
- ▶ Αποεπιλογή Fixed Duration
- ▶ Transition Duration = 0

Ρυθμίσεις animation

- ▶ Η ιδέα είναι ότι πατώντας το αριστερά Ctrl, ο Player θα σέρνεται και μετά θα επιστρέφει στο Run (αυτό είναι ήδη ρυθμισμένο γιατί στο transition slide→run έχουμε επιλεγμένο το Has exit time)
- ▶ Αφαιρούμε το Loop Time από το animation slide. Με επιλεγμένο το animation από το φάκελο animations, βρισκω την επιλογή στον Inspector.



Κώδικας

1. Προσθέτουμε μια μεταβλητή bool slide → private bool slide;
2. Στη Reset function που έχουμε δημιουργήσει από προηγούμενα, θα πάρουμε και θα την ορίσουμε σε false → slide=false;
3. Τώρα θα πρέπει να δώσουμε το έναυσμα για να γίνει true. Θα το κάνουμε μέσα στη handleMovement

```
if (slide=true  
&&!this.myAnimator.GetCurrentAnimatorStateInfo(0).IsName ("Slide"))  
{ myAnimator.SetBool("slide", true)  
}  
  
else if (!this.myAnimator.GetCurrentAnimatorStateInfo  
(0).IsName("slide"))  
{ myAnimator.SetBool("slide", false);  
}
```

αν το slide είναι true ΚΑΙ αν δεν παίζει ήδη το animator slide

αν δεν είναι slide το όνομα του animation

αν δεν παίζει το slide, να γυρνάει και το bool slide σε false

Κώδικας

4. Τέλος θα προσθέσουμε το πλήκτρο με το οποίο δίνεται το έναυσμα στη HandleInput

```
if(Input.GetKeyDown(KeyCode.LeftControl))  
{  
    slide=true;  
}
```

ΔΟΚΙΜΑΖΟΥΜΕ!

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 slide;
    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();
        ResetValues();
    }
}
```

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

    if (slide &&
!this.myAnimator.GetCurrentAnimatorStateInfo(0).IsName("Slide"))
    {
        myAnimator.SetBool("slide", true);
    }
    else if
(!this.myAnimator.GetCurrentAnimatorStateInfo(0).IsName("Slide"))
    {
        myAnimator.SetBool("slide", false);
    }

    myAnimator.SetFloat("speed", Mathf.Abs(horizontal));
}
private void HandleAttacks()
{
    if(attack &&
!this.myAnimator.GetCurrentAnimatorStateInfo(0).IsTag("Attack"))
    {
        myAnimator.SetTrigger("attack");
        myRigidbody.velocity = Vector2.zero;
    }
}
```

```
private void HandleInput()
{
    if(Input.GetKeyDown(KeyCode.LeftShift))
    {
        attack = true;
    }
    if(Input.GetKeyDown(KeyCode.LeftControl))
    {
        slide = 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;
    }
}
private void ResetValues()
{
    attack = false;
    slide = false;
}
}
```

Διόρθωση κίνησης

- ▶ Τώρα πηγαίνει πέρα δώσε τσουλώντας κάτι που δε θέλουμε. Θέλουμε να τσουλάει προς μια κατεύθυνση και να μπορεί να γυρίσει προς την άλλη αφού έχει τελειώσει αυτή η κίνηση
- ▶ Προσθέτουμε στον κώδικα το παρακάτω

```
private void HandleMovement(float horizontal)
{
    if(!myAnimator.SetBool("slide") &&
!this.myAnimator.GetCurrentAnimatorStateInfo(0).IsTag("Attack"))
    {
        myRigidbody.velocity = new Vector2(horizontal * movementSpeed,
myRigidbody.velocity.y);
    }
}
```

Τώρα του λέμε ότι όταν το slide είναι true μπορεί να κυλήσει αλλά του λέμε επίσης ότι όταν κάνει slide δε μπορεί να κουνιέται